

Copie riservate agli abbonati

ISSN 0392-8996

PERSONAL SOFTWARE

ANNO 5 N. 37
MARZO 1986
L. 4.500

LA PRIMA RIVISTA EUROPEA DI SOFTWARE PER PERSONAL COMPUTER

Spedizione in abbonamento postale gruppo III/70



GRUPPO EDITORIALE
JACKSON
DIVISIONE PERIODICI

GRAFICA 3D PER C 16



CRUCI MSX

MUSIC UTILITY C 64

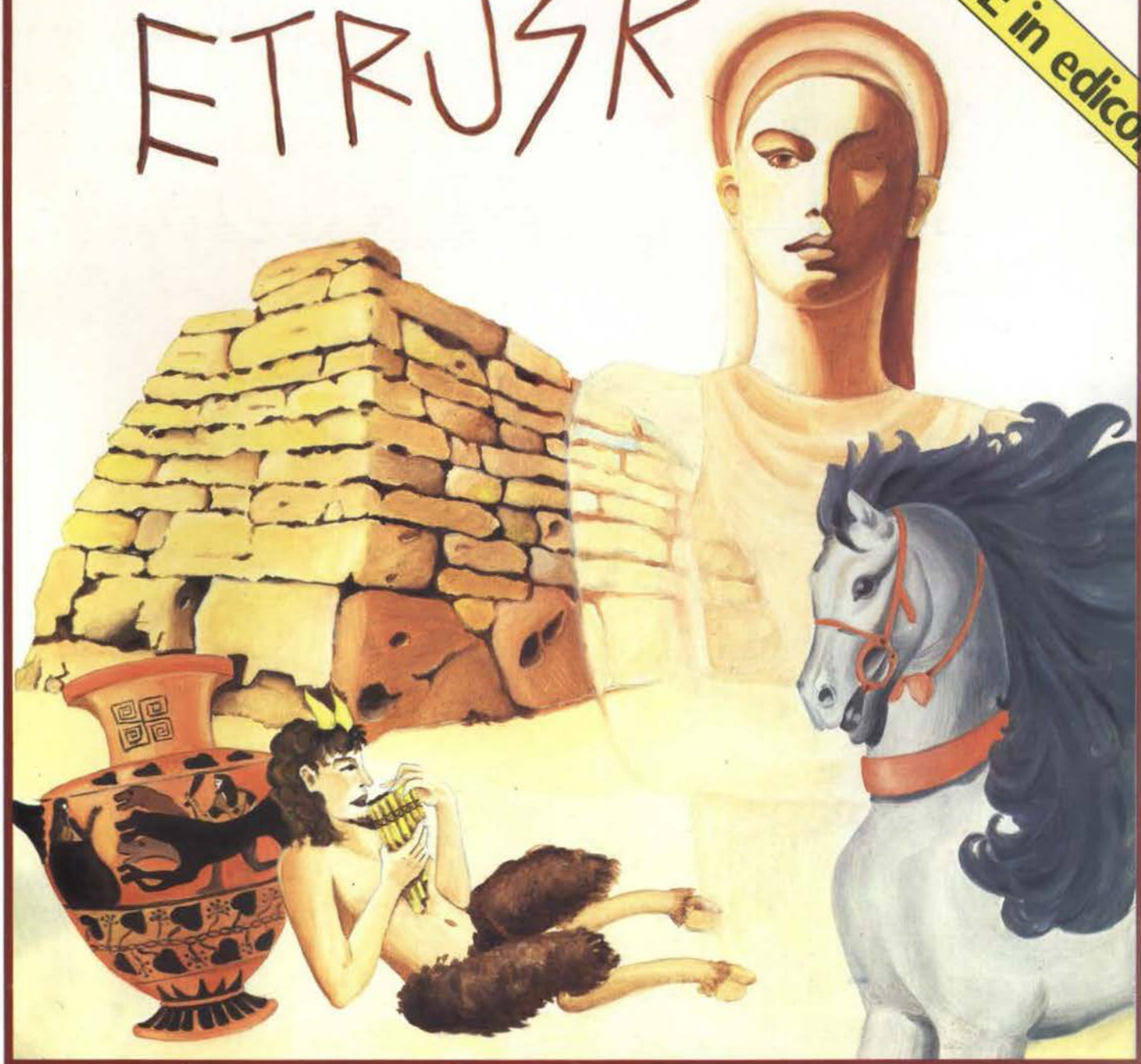
PRO.DOTTO PER APPLE

PANICO! PER SPECTRUM

TAPPETO MOBILE PER APPLE

ETRUSK

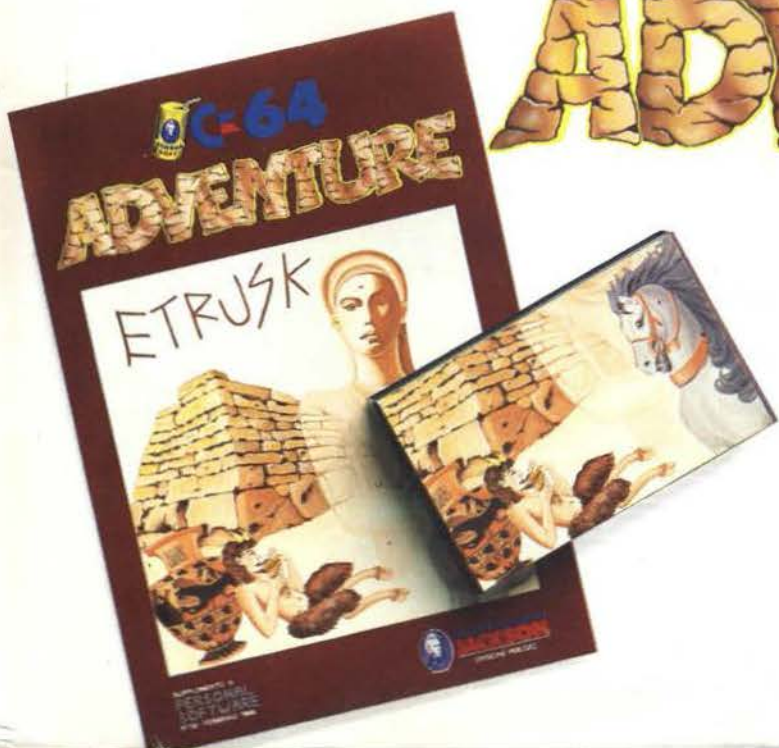
È in edicola



ADVENTURE



C-64



Entra anche tu nel misterioso mondo degli Etruschi con una favolosa avventura interamente italiana. Incontrerai fantasmi, regine affascinanti, lupi e altre strane creature che popolano le antichissime mura di Veio, autentica città etrusca. Scopri con un gioco irresistibile i misteri di questa affascinante civiltà.



GRUPPO EDITORIALE
JACKSON

DIVISIONE PERIODICI

SOMMARIO



Questo mese:
Grafica 3D
per Commodore 16.

ANNO 5
N. 37
MARZO 1986

PRO.DOTTO 2° <i>di Luca Accomazzi</i>	APPLE II	7
GRAFICA A TRE DIMENSIONI <i>di Valerio Anselmo</i>	COMMODORE 16	17
PANICO! <i>di Marcello Spero</i>	ZX SPECTRUM	21
EMEROTECA CON "ARCHIVE" <i>di Giovanni Tisi</i>	SINCLAIR QL	27
CRUCI MSX <i>di Luciano Cassi</i>	MSX	29
UN PO' D'ORDINE FRA I NOSTRI DATI <i>di Luca Zaninello</i>	CASIO PB-100-200-300	35
TAPPETO MOBILE 2° <i>di Ugo Rossini</i>	APPLE II	41
MUSIC UTILITY <i>di Tommaso Palermo</i>	COMMODORE 64	51

RUBRICHE

PERSONAL NEWS <i>a cura di Marco Giacobazzi</i>		4
I SEGRETI DEI PERSONAL: Input function <i>di Sergio Borsani</i>	TEXAS TI99/4A	56
La funzione Print Using <i>di Tullio Policastro</i>	ZX SPECTRUM	58
L'uso delle unità a dischi Commodore <i>di Enrico Martelli</i>	COMMODORE 64	59
PERSONAL MARKET		65

Sempre più facile

La Electronic Service annuncia una nuova versione dell'espansione Es-9 per Commodore 64 comprendente modifiche al Fast disk loader, estensioni di comandi eseguibili da Basic e introduzione di abbreviazioni per agire da programma su disco. È stato altresì modificato il comando List directory (6) con l'aggiunta delle possibilità di cancellazione e cambiamento del nome dei file contenuti sul floppy. Per il Basic è stato introdotto il comando \$ mediante il quale è possibile ottenere la lista del contenuto di un disco con tutte le modalità e possibilità del comando List directory (caricamento, cancellazione, modifica del nome).

*Electronic Service di Loris Massarelli
Via Prinetti 14/16
Tel. 050-574563
56100 Pisa*

La stagione di Commodore Time

Sulla scia del successo e della fama conquistata producendo e distribuendo home e personal computer, la Commodore si rivolge ad altri settori commerciali, sempre attinenti ai consumi dei giovanissimi.

Per i bambini la collezione Commodore Time propone orologi al quarzo disponibili in quattro colori, impermeabili fino a 30 metri.

La serie ha un prezzo di 30 mila lire al pubblico.

Ai ragazzi viene proposta una linea elettronica con quadrante analogico e lancette colorate, cinturino e quadrante in acciaio lucido/satinato.

Il tutto per 85 mila lire.

Opinione

*Via G. Giacomo Mora, 22
20125 Milano
Tel. 02-8373081*

Interfaccia per Apple II

La Deltaacque Elettronica ha messo a punto una scheda di interfaccia per Apple II e compatibili, adatta per una vasta gamma di applicazioni quali acquisizione dati, controllo di processo e didattica. In un'unica scheda sono presenti un convertitore analogico/digitale a otto bit con 16 ingressi multiplexati, un Via 6522 con 16 e più linee di I/O programmabili, due timer/counter a 16 bit e uno shift register, due convertitori D/A a otto bit e otto linee d'ingresso Ttl. La I/O Card viene fornita con un manuale d'uso in italiano completo di schemi e costa 300.000 lire più Iva.

*Deltaacque Elettronica
Via dei Gelsi, 35
50040 Calenzano (FI)
Tel. 055-882191*





Apricot per tutti

Agili, aggressivi e poco ingombranti, arrivano anche in Italia i personal computer della Apricot, distribuiti dalla Ditron di Milano. Questo sbarco ha le carte in regola per essere ritenuto importante: la Apricot International, primo gruppo inglese di microcomputer e numero quattro in Europa, ha dichiarato di volersi inserire tra le prime cinque marche nella gamma dei business computer in Italia nel 1986. Spera perciò che la collaborazione con la Ditron e con gli altri subdistributori regionali dia a breve termine i frutti attesi, facendo leva su un parco macchine ampio e innovativo. Si tratta di sistemi Ms-Dos, dotati di drive per floppy da 3,5", tastiera e mouse a raggi infrarossi, equipaggiati con la Gem collection e altro software di emulazione del Pc Ibm. La serie Apricot Collection, con i modelli F2, F10 e Fp, punta sul mondo dei computer professionali e da ufficio, con macchine dotate di 512 Kbyte di Ram e

almeno un floppy capace di 720 Kbyte. Il portatile Fp monta uno schermo Lcd da 25 righe con una risoluzione di 640 per 200 punti ed è collegabile a un monitor a colori, gestiti da 128 Kbyte supplementari di Ram. Costa 3.700.000 lire più Iva. I modelli da scrivania sono l'F2, con due floppy per un totale di 1.440 Kbyte in linea, e l'F10 con Winchester da 10 Mbyte incorporato. I prezzi sono rispettivamente di 4.720.000 e 6.675.000 lire senza Iva.

*Ditron S.p.A.
Viale Certosa, 138
20156 Milano
Tel. 02-3085645*

Un po' di tutto...

L'esposizione universale di Vancouver aprirà i battenti tra poco più di due mesi. Expo '86 si svolgerà infatti in

Canada dal 2 Maggio al 13 Ottobre 1986. - Expo Info - P. O. Box 1800 - Vancouver B. C. Canada

● L'esposizione *Comdex/Europe* avrà luogo quest'anno a Nizza dal 10 al 12 Giugno 1986. Parallelamente a essa si svolgerà una conferenza sul business software rivolta espressamente a distributori e rivenditori. The Interface Group - Wtc Strawinskylaan 1245 - 1077 XX Amsterdam - The Netherlands - Tel. (31) 20-621941.

● Disponibile su richiesta il catalogo aggiornata *Evm Computer* per la vendita per corrispondenza di materiale per i micro Commodore. Con il numero più recente è stata introdotta la commercializzazione di prodotti messi a punto dagli utenti. Volete essere della partita? - Evm Computer - Via Marconi 9/A - 52025 Montevarchi (AR) - Tel. 055-980242.



Commutatore per periferiche

Per risolvere i problemi legati all'utilizzo di diverse periferiche da connettere a un'unica uscita, la Brm propone il suo Transfer switch, commutatore manuale che dispone di un input e di due o più output. Le interfacce previste sono la seriale Rs-232C e la parallela Centronics. È possibile pertanto collegare un singolo Pc a due o più stampanti o modem, oppure commutare due o più Pc su una risorsa comune (modem, stampante, plotter). Esiste anche una versione con due input e due output di questi componenti passivi che non richiedono manutenzione. Il tutto in 18 per 5,5 per 13 cm con un peso di 1,3 kg. Il prezzo varia dalle 200 alle 300 mila lire secondo il tipo di interfaccia richiesto e il numero di



Dalla grande edicola Jackson

Tutto sul personal computer

PC

L'unica rivista italiana dedicata ai sistemi MS-DOS, Personal computer IBM e compatibili.
11 numeri all'anno: L. 5.000 a numero
Abbonamento: solo L. 44.000

Personal

L'unica rivista indipendente per gli utenti dei personal computer Olivetti.
11 numeri all'anno: L. 5.000 a numero
Abbonamento: solo L. 42.000

COMPUSCUOLA

La rivista di informatica nella didattica per la scuola italiana.
9 numeri all'anno: L. 3.000 a numero
Abbonamento: solo L. 20.000

medical computer

Una novità assoluta per l'editoria tecnico-scientifica, studiata appositamente per i medici, la Rivista di personal computer, affari personali e tempo libero.
11 numeri all'anno: L. 5.000 a numero
Abbonamento: solo L. 45.000

Bit

La prima rivista europea di personal computer, software e accessori. Con test, novità, analisi del mercato ...
11 numeri all'anno: L. 5.000 a numero
Abbonamento: solo L. 43.000

Quando l'informazione fa testo

In busta chiusa inviate questo coupon a:
Gruppo Editoriale Jackson
via Rosellini, 12 - 20124 MI

Desidero ricevere GRATIS un numero della Rivista _____

(allego L. 1.000 in francobolli per contributo spese di spedizione)

Inviatemi GRATIS il Catalogo della Biblioteca JACKSON (allego L. 1.000 in francobolli per contributo spese di spedizione)

Nome _____

Cognome _____

via _____

CAP _____ Città _____

PERSONAL NEWS

uscite desiderato.

Brm Italiana
Lungo Dora Firenze, 21 bis
10152 Torino
Tel. 011-287884

Corsi per genitori (e figli)

La Kinder Computer School ha organizzato una serie di corsi di introduzione alla programmazione in Basic per i modelli Commodore 64 e 128. Oltre ai tradizionali corsi per ragazzi, la scuola propone una serie di 20 lezioni per genitori e figli, allo stesso prezzo (220 mila lire) del corso singolo. Adulto e ragazzo utilizzeranno lo stesso materiale (computer e dispense), partecipando insieme alle lezioni. Sono previste versioni dei corsi da effettuarsi presso scuole elementari, medie e superiori.

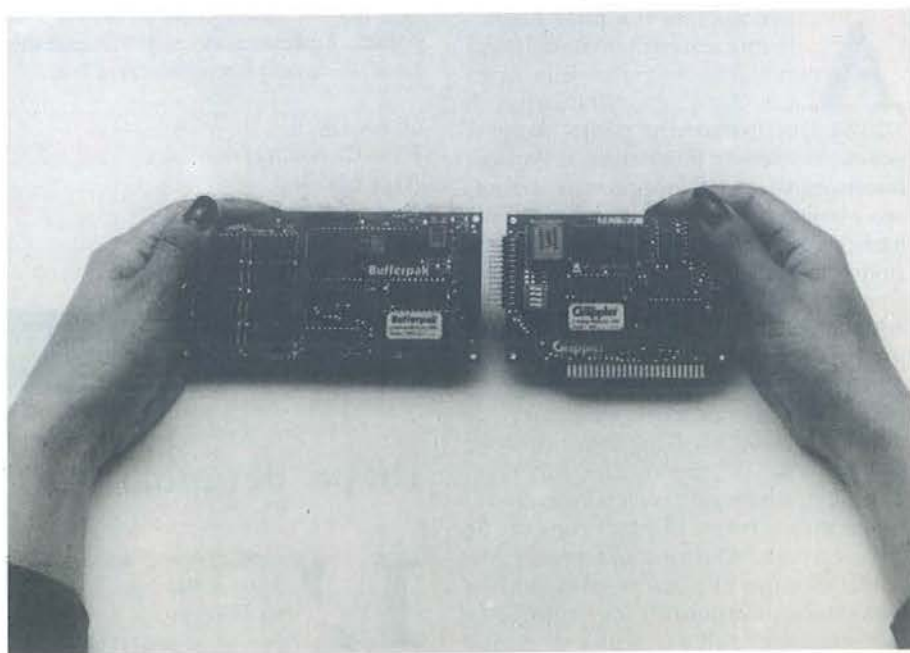
Kinder Computer School
Via Vittoria Colonna, 2
20149 Milano
Tel. 02-4815240

Interfaccia per Apple

Orange Micro annuncia la nuova versione seriale dell'interfaccia Grappler printer per collegare le stampanti Imagewriter I e II e la Scribe all'Apple IIe.

La Serial Grappler+ offre agli utenti delle Image Writer un insieme di caratteristiche non sempre presenti nei prodotti della concorrenza. È possibile ottenere la copia dello schermo in bianco e nero o a colori, richiedere grafici invertiti, ruotati e con diverse deformazioni (enhanced ed enlarged), oltre a disegni in doppia risoluzione. L'opzione Bufferpack, che si innesta direttamente nella scheda Grappler+ senza necessità di cavi né di alimentazione propria, fornisce 16, 32 o 64 Kbyte di buffer per la stampante, sollevando il sistema dall'attesa del completamento delle operazioni. Orange Micro sta sviluppando altri prodotti opzionali in stretta collaborazione con i progettisti della Apple.

Orange Micro Inc.
1400 N. Lakeview Ave.
Anaheim, CA 92807 Usa
Tel. (714) 7792772



In questa puntata trattiamo il principale componente del Pro.dotto: un programma che, interagendo direttamente con ProDos, vi permette d'intervenire direttamente nella struttura dei dischi. Lo abbiamo chiamato...

L'editor di blocchi

Un blocco è l'unità minima di spazio sul disco che ProDos può indirizzare separatamente, ed è composta da 512 byte (mezzo Kbyte). Normalmente voi non vi occupate di come i blocchi sono disposti sul disco e di come vengono uniti a formare i file: ProDos e Basic.System (il programma che fa interagire ProDos e Applesoft) se ne occupano per voi.

Vi sono però dei momenti in cui diventa utile, se non indispensabile, poter leggere e scrivere i blocchi del disco: perché parte del disco è stata distrutta e volete recuperare almeno parte dei dati che conteneva: oppure perché qualche programma di sistema ha combinato un pasticcio con il vostro disco rovinandone il catalogo: o ancora per studiare un programma nella sua forma sul disco, prima che possa proteggersi una volta caricato in memoria. A questo scopo è stato creato l'editor di blocchi.

Editor di blocchi, come tutti i programmi di Pro.dotto, è stato ottimizzato nella lunghezza: occupa solo 2 Kbyte della memoria. Tuttavia, potrebbe darsi che voi dobbiate conservare una determinata zona di memoria per studiarla (come vedremo l'editor di blocchi permette di esaminare e modificare anche la memoria di Apple II oltre che i dischi). Per questo motivo abbiamo predisposto tre differenti versioni di editor di blocchi, chiamate:

- Edit. System;
- Edit. 9200;
- Edit. 800.

Per motivi di spazio, pubblichiamo ad accompagnamento di questo articolo il listato del solo Edit.9200; potete trovare gli altri due programmi nel dischetto Pro.dotto che potete ordinare alla redazione (vedere *Servizio software*). I tre programmi sono quasi identici; discuteremo in seguito delle loro differenze, mentre ora ne spieghiamo gli aspetti comuni.

Pro.dotto

Proseguiamo con la descrizione dell'editor di blocchi

di Luca Accomazzi

Parte seconda

Normalmente dovrete usare Edit.9200 per lavorare, e riservare le altre due versioni per scopi speciali. Per le prime prove, usate un disco protetto in scrittura o un disco sacrificabile, in modo che eventuali errori non portino guasti irreparabili. Riservate le operazioni complesse a quando vi sarete familiarizzati con il programma.

Se per caso sull'Apple II non è presente la scheda 80 colonne, indispensabile all'editor di blocchi, questo presenta la scritta "Manca la 80 col!" e non può essere usato. Sia una scheda 80 colonne normale sia la versione con espansione di memoria sono adatte all'uso. Apple IIc possiede una scheda 80 colonne incorporata: per le versioni precedenti di Apple II è necessario acquistarla separatamente dal vostro rivenditore Apple.

Una volta caricato in memoria e lanciato, editor di blocchi presenta alcune righe di presentazione e vi chiede di premere un tasto per cominciare il lavoro. Quando avrete premuto un qualsiasi tasto, verrà attivata la scheda 80 colonne e vedrete una videata di numeri, separati in quadranti per renderla più leggibile, sormontata da alcune righe di spiegazione. Noterete che il display comprende esattamente 512 byte, un blocco appunto, numerati da 000 a 1FF in esadecimale, e divisi in righe di 32 (\$20).

I comandi di editor di blocchi

Tutti i comandi sono costituiti da una sola lettera, che viene accettata solo se premuta maiuscola (sugli Apple IIe e Apple IIc varrà la pena di mantenere abbassato il tasto maiuscole Caps-

Lock durante l'uso dell'editor). Nel caso premiate un tasto che non corrisponde ad alcun comando, verrà emesso un "buzz" e apparirà nello spazio messaggi (in alto a destra sulla seconda riga) il messaggio "Opzione errata". Potete provare battendo la sbarra spaziatrice: non accadrà nulla di male.

A (Ascii): il comando A modifica il display per mostrare i codici Ascii dei byte anziché i loro valori numerici. Usato una seconda volta commuta nuovamente ai valori numerici e così via.

B (blocco): con il comando B è possibile scegliere il numero logico di un blocco del disco da leggere o scrivere. Il numero dev'essere composto di tre cifre, ognuna da 0 a F: editor di blocchi vi permette infatti di operare su dischi sino a 2 Mbyte di memoria, e dunque potete usarlo sia per il Disk II, che per il nuovo Unidisk da 800 Kbyte, che per il ProFile. Ricordate nell'uso normale che un dischetto standard da 5,25" contiene solo 280 blocchi (in esadecimale numerati da 000 a 117), e ogni tentativo di leggere blocchi oltre questo numero produrrà un "Errore di I/O", mentre il tentativo di scrivere vi darà l'errore "Protetto in scrittura".

Il disco Ram (vedi sotto l'opzione S) ha 120 blocchi numerati in esadecimale da 000 a 077.

L'uso del comando B non provoca né la lettura né la scrittura del blocco in questione, ma preseleziona solamente quel blocco particolare. Il valore cui è fissato inizialmente il numero di blocco è 000.

D (disassemblaggio): con D otterrete il disassemblato dei byte nel blocco; se conoscete il linguaggio macchina sapete quanto questa possibilità sia fruttuosa per esaminare i programmi. Il di-

sassemblato viene visualizzato a gruppi di 20 istruzioni per volta; tra i gruppi il computer presenta il cursore e attende la pressione di un tasto.

Verranno disassemblati i 512 byte del blocco: tuttavia, se premete Control-C durante una delle pause, l'operazione viene interrotta e tornerete al display normale. Se per qualche motivo avrete bisogno di bloccare lo scorrere del disassemblato oltre agli stop fissati ogni 20 righe, potete ottenerlo premendo Control-S. Ogni tasto, poi, farà riprendere le operazioni.

E (editare la memoria): con E potete modificare uno o più dei byte nel blocco. Vi verrà dapprima chiesto il numero del byte da modificare (da 000 a 1FF: se scegliete un numero al di fuori di questo

raggio l'editor di blocchi risponde con un messaggio d'errore). L'editor risponderà mostrando il messaggio "da ? a", dove ? è il valore attuale del byte, espresso in esadecimale o in codice Ascii, a seconda della scelta attuale, compiuto con l'opzione A. Poi verrà presentato il cursore per l'introduzione del nuovo valore. Dopo aver battuto il valore modificato cui volete porre il byte potete scegliere fra tre possibilità.

Se batterete Return il valore introdotto verrà sostituito al valore che si trovava in memoria, e terminerà l'opzione Editare.

Se batterete Sbarra il cambiamento verrà effettuato e vi verrà chiesto automaticamente un nuovo valore per il byte seguente: questa operazione è utile per

modificare più valori consecutivi, magari per cambiare un messaggio.

Se batterete Esc la modifica non verrà effettuata, e l'opzione Editare verrà terminata.

La modifica avviene comunque solo in memoria: per apportarla anche al blocco sul dischetto occorre scrivere il blocco modificato con "AW".

F (floppy): l'opzione F seleziona alternativamente il drive uno o due rispetto allo slot selezionato. Sull'Apple IIc il drive uno è il drive interno, il drive due è quello esterno, se esiste. Tentare di operare con un drive che non esiste produce l'errore "Non è un drive".

H (high bit): i messaggi alfanumerici vengono custoditi in memoria secondo una corrispondenza con i numeri interi chiamata codice Ascii. Il codice assegna una corrispondenza biunivoca tra ciascun numero e una lettera, numero o simbolo; ciascun byte in memoria tipicamente custodisce l'equivalente Ascii di un carattere. Ma i caratteri Ascii sono 128, mentre un byte può valere da 0 a 255. Accade così che i valori da 128 a 255 ripetono i medesimi caratteri dei valori 0 - 127. Quando volete alterare un messaggio, una scritta, una stringa di caratteri è necessario scegliere tra una delle due possibilità: usare i valori 0-127 (bit alto non settato, cioè = 0) o usare i valori 128-255 (bit alto settato, cioè = 1). Una scelta sbagliata può portare a una visualizzazione abnorme, tipicamente in inverse o flash anziché in caratteri normali. Con l'opzione H si seleziona il bit alto a zero o a uno. Potete scegliere osservando il codice dei caratteri che dovete sostituire: se i valori sono superiori a 80 - esadecimale - allora dovrete assicurarvi di avere il bit alto posto a uno. L'uso dell'opzione H visualizza un messaggio che chiarisce a che valore viene posto il bit alto nello spazio messaggi. Per motivi di spazio, l'opzione H è disponibile solo in Edit.9200.

M (memoria): con l'opzione M si seleziona una zona di memoria differente da quella standard come buffer per le operazioni e per lavoro. Va indicato un numero di pagina da 00 a FE, che sia pari. *Attenzione:* non bisognerebbe mai usare come buffer le pagine da 00 a 07 e da 92 in su, utilizzate dal sistema. ProDos impedirebbe comunque di leggervi qualunque blocco, ma l'uso dissennato dell'opzione E per modificare alcuni va-

Listato 1 - Il modulo Loader che riloca l'editor di blocchi.

```

SOURCE   FILE #01 =>LOADER
0000:      1 *
0000:      2 * Edit.9200 Loader
0000:      3 *
2000:      4      ORG    $2000

2000:      5      LST    ON,GEN,NOV,NOA
2000:      BE0C 6 PRINTERR EQU $BE0C
2000:      BE00 7 WARMSTRT EQU $BE00
2000:      BE03 8 DOSCMD  EQU $BE03
2000:      0200 9 SYSBUF  EQU $0200
2000:      10 ;
2000:A2 00      11      LDX    #00
2002:BD 44 20      12 CLOSE  LDA    DOS,X      ;Chiede a BSYS un CLOSE
2005:9D 00 02      13      STA    SYSBUF,X    ; per evitare problemi
2008:EB           14      INX
                        ; con HIMEM
2009:E0 06      15      CPX    #6
200B:DO F5 2002  16      BNE    CLOSE
200D:20 03 BE      17      JSR    DOSCMD
2010:C9 00      18      CMP    #0
2012:F0 06 201A  19      BEQ    INIT      ;If acc <> 0 then error
2014:20 0C BE      20      JSR    PRINTERR
2017:4C 00 BE      21      JMP    WARMSTRT    ;Esce senza sistemarsi
201A:A9 00      22 INIT   LDA    #00
201C:B5 FE      23      STA    $FE
201E:A9 92      24      LDA    #$92
2020:B5 FF      25      STA    $FF      ;TD
2022:A9 21      26      LDA    #$21
2024:BE FD      27      STA    $FD
2026:A9 00      28      LDA    #00      ;FROM
2028:B5 FC      29      STA    $FC
202A:AB           30      TAY
202B:B1 FC      31 COPYSELF LDA ($FC),Y
202D:91 FE      32      STA ($FE),Y
202F:E6 FC      33      INC    $FC
2031:D0 02 2035  34      BNE    C1
2033:E6 FD      35      INC    $FD
2035:E6 FE      36 C1    INC    $FE
2037:D0 02 203B  37      BNE    C2
2039:E6 FF      38      INC    $FF
203B:A5 FF      39 C2    LDA    $FF
203D:C9 9A      40      CMP    #$9A
203F:DO EA 202B  41      BNE    COPYSELF    ;Se copiato tutto, esci
2041:4C 00 92      42      JMP    $9200      ;Vai!
2044:           43      MSB    ON
2044:C3 CC CF D3  44 DOS   ASC    "CLOSE"
2049:C5           45      DFB    $BD

** SUCCESSFUL ASSEMBLY := NO ERRORS
** ASSEMBLER CREATED ON 15-JAN-84 21:28
** TOTAL LINES ASSEMBLED 45
** FREE SPACE PAGE COUNT 84

```


lori potrebbe causare guasti imprevedibili nel ProDos stesso. È possibile selezionare quelle pagine per renderne possibile la lettura, ma non andrebbero mai modificate. Il tentativo di selezionare la pagina CO produce il messaggio "Buffer Illegale". La pagina CO, infatti, non è una normale pagina di memoria e pertanto non può venire né letta né scritta dissennatamente.

R (read): con R viene letto dal disco, nella pagina di memoria selezionata, il blocco selezionato.

S (slot): serve a selezionare lo slot del disco su cui lavorare: poiché il disco può solo trovarsi in uno slot compreso tra l'uno e il sette: vengono accettati solo quei valori, e un numero differente provoca il messaggio d'errore "Indicare 1,2...7".

Tentare di operare con un drive che non esiste produce l'errore "Non è un drive". Il disco Ram degli Apple con 128 Kbyte è indicato come slot tre drive due.

Selezionando quei valori con le opzioni S e F, siete in grado di esaminare anche la memoria aggiuntiva del vostro Apple (per la memoria ordinaria, i 64 Kbyte principali, usate l'opzione M).

Control-W (write): con questo comando il buffer (i 512 byte attualmente visualizzati) viene copiato sul disco nel blocco indicato nella seconda riga. È necessario premere anche il tasto Control, oltre al tasto W: in questo modo si rende impossibile scrivere per errore un blocco sul disco. Scrivere dissennatamente sul disco è un'operazione pericolosa che rende quasi certamente il disco inservibile, illeggibile per ProDos e dunque perdendo i dati contenuti. Modificate solo dei back-up dei vostri dischi, e accertatevi sempre di sapere quel che state facendo.

X (exit): questo comando è disponibile solo in Edit.System, e viene utilizzato per uscire dall'editor di blocchi e caricare Basic.System.

Edit.System è un programma di sistema, e dunque può venire usato anche senza Basic.System (nulla impedisce di usarli entrambi). Questa possibilità si rivela utilissima quando occorre studiare un programma in memoria prima del caricamento di ProDos: il lancio di Basic.System cancellerebbe buona parte della memoria, mentre usando un disco che non lo contiene, ma che dispone di Edit.System, permette di mantenere il più possibile immutata la memoria di Apple II.

Notate che Edit. 9200 ed Edit. 800 non sono programmi di sistema, ma semplici programmi binari (tipo Bin) che possono venire usati e caricati in memoria solo con Basic.System.

Normalmente, un programma di sistema (i programmi con il nome che termina in .System e indicati nel catalogo del disco come tipo Sys) dovrebbe poter terminare dando il controllo a un altro qualsiasi programma di sistema. Tutta-

Listato 2 - Il programma editor di blocchi

```

SOURCE FILE #01 =>ED
0000:
0000: 1 #
0000: 2 # programma PRODOS BLOCK EDITOR
0000: 3 # usa BLOCKREAD e BLOCKWRITE ProDOS, Applesoft, Monitor
0000: 4 # Creato 4 Apr 1985 by MisterAkko
0000: 5 # Final assembly 12 May 1985
0000: 6 #
0000: 7 # Versione in alta memoria: si colloca in $9200
0000: 8 #
0000:
0000: 9 LST DN,NOA,GEN
0000: 10 MLICAL EQU $BF00
0000: 11 PAG EQU $06
0000: 12 HOME EQU $FC5B
0000: 13 TEXT EQU $FB39
0000: 14 SCRN EQU $409
0000: 15 KEYPRS EQU $FDOC
0000: 16 EIGHTY EQU $C300
0000: 17 MACHID EQU $BF98
0000: 18 COUT EQU $FDED
0000: 19 CROUT EQU $DAFB
0000: 20 HTAB EQU $57B
0000: 21 VPOS EQU $25
0000: 22 VTAB EQU $FC22
0000: 23 INCHR EQU $D553
0000: 24 SPKR EQU $C030
0000: 25 FBYTE EQU $FDDA
0000: 26 PRHEX EQU $FDE3
0000: 27 WAIT EQU $FC48
0000: 28 PR$BLNK EQU $F94B
0000: 29 SETINV EQU $FE80
0000: 30 SETNORH EQU $FEB4
0000: 31 here EQU $9200
0000: 32 #
0000: 33 # Program beginning: clear screen and present title
0000: 34 #
0000: 35 ORG here
0000: 36 JSR TEXT
0000: 37 JSR HOME
0000: 38 LDA #0
0000: 39 LDA STR,X
0000: 40 STA SCRN,X
0000: 41 INX
0000: 42 CPX #19
0000: 43 BNE TITLE
0000: 44 LDY #0
0000: 45 LDA STR2,X
0000: 46 STA SCRN+$80,X
0000: 47 INX
0000: 48 CPX #19
0000: 49 BNE AKKO
0000: 50 LDX #0
0000: 51 # Controlla se è presente la ottanta colonna
0000: 52 LDA MACHID ; Controlla se bit 1=1
0000: 53 AND #2
0000: 54 BNE READY
0000: 55 LDA STR4,X
0000: 56 STA $600,X
0000: 57 INX
0000: 58 CPX #17
0000: 59 BNE NOB0
0000: 60 JMP QUIT
0000: 61 LDA STR3,X
0000: 62 STA $607,X
0000: 63 INX
0000: 64 CPX #1D
0000: 65 BNE ready
0000: 66 LDA VPOS
0000: 67 STA VPOS
0000: 68 JSR VTAB
0000: 69 LDA #20
0000: 70 STA HTAB

```

via la routine usata dall'Apple Computer per realizzare questa possibilità occupa da sola circa 5 Kbyte: abbiamo scelto un'uscita più limitata, ma più snella.

Con X Edit.System cerca di caricare e lanciare Basic.System dal disco nel drive uno dello slot sei. Sono possibili gli errori "Manca Basic.Sys" ed "Errore di I/O". Se per qualche strano motivo Basic.System fosse presente sul disco, ma in forma danneggiata, è possibile che editor di blocchi venga distrutto, ma Basic.System non sia caricato: in questo caso verrà mostrato il messaggio "Errore irrecuperabile" e il computer si fermerà in Monitor. Quando ciò accadesse dovrete rilanciare interamente il sistema con Control Mela-vuota Reset o spegnendo e riaccendendo.

Esc (uscita dal programma): con Esc si esce dal programma editor di blocchi per Basic.

Differenza: da Edit.800 Esc termina irre-

vocabilmente il programma. Da Edit.9200 Esc esce in Basic, ma editor di blocchi resta in memoria, sempre disponibile (questo ruba 2 Kbyte ai vostri programmi Basic: se volete eliminare dalla memoria l'editor potete farlo usando il programma FP di Pro.dotto pubblicato nello scorso numero, oppure rilanciando Basic.System col comando Basic.System). Potete rientrare nell'editor di blocchi con il comando & (ampersand). Battendo & <Return> vi lascerà nuovamente in Edit. 9200.

Da Edit.System Esc funziona esattamente come con Edit.800; ma se Basic.System non è presente in memoria, poiché ProDos ha caricato direttamente l'editor, otterrete il messaggio d'errore "Manca Basic.Sys" e dovrete usare il comando X.

Mela vuota?: questo comando, come probabilmente saprete, dovrebbe fornirvi un aiuto visualizzando un messaggio che chiarisca il funzionamento del

programma. Tuttavia, per mantenere molto ridotte le dimensioni del programma, non è stato possibile implementare esaurienti messaggi di spiegazione, che occupano sempre molta memoria. Premendo Mela vuota e "?" dunque viene semplicemente visualizzato il messaggio "Consulta il manuale".

← e → (*freccia a sinistra e a destra*): le due frecce rispettivamente decrementano e incrementano di uno il numero di blocco selezionato, ma non leggono né scrivono il blocco dal disco. Lo scopo è una efficiente selezione del blocco prescelto.

↑ e ↓ (*freccia in alto e in basso*): hanno lo stesso effetto di freccia a sinistra e a destra, ma, contemporaneamente a incrementare e decrementare il numero del blocco selezionato, leggono il blocco stesso in memoria. In questo modo è molto semplice scorrere i contenuti di un intero dischetto sul video (mantenendo premuto il tasto freccia in alto).

Seguito listato editor di blocchi

924E:14C 0B 94	71	JMP	CONT						
9251:	72 *								
9251:	73 *	STUFF							
9251:	74 *								
9251:	75 *								
9251:C2 EC EF E3	76 MSB	DN							
Buffer 0800	76 MSB1	ASC	'Blocco						
9255:ES EF AO CE	76 MSB1	ASC	'Block						
9259:AE AO BO BO			editor 1,0.						
925D:BO AO AD AO									
9261:C4 F2 E9 F6									
9265:E5 AO B1 AO									
9269:AD AO D3 EC									
926D:EF F4 AO B6									
9271:AO AD AO C2									
9275:F5 E6 E6 E5									
9279:F2 AO BO B8									
927D:BO BO AO AD									
9281:AO DO F2 EF									
9285:C4 CF D3 AF									
9289:DO E1 F3 E3									
928D:E1 EC AO C2									
9291:EC EF E3 EB									
9295:AO E5 E4 E9									
9299:F4 EF F2 AO									
929D:B1 BE BO									
92A0:AO BO BO AD	77 MSB3	ASC	180-1A0-1C0-1ED-						
AO-C0-E0			100-120-140-160-						
92A4:AO B2 BO AD									
92A8:AO B4 BO AD									
92AC:AO B6 BO AD									
92B0:AO AO AO AD									
92B4:AO B8 BO AD									
92B8:AO C1 BO AD									
92BC:AO C3 BO AD									
92C0:AO C5 BO AD									
92C4:AO AO AO AD									
92C8:B1 BO BO AD									
92CC:B1 B2 BO AD									
92D0:B1 B4 BO AD									
92D4:B1 B6 BO AD									
92D8:AO AO AO AO									
92DC:B1 B8 BO AD									
92E0:B1 C1 BO AD									
92E4:B1 C3 BO AD									
92E8:B1 C5 CF AD									
92EC:C5 C4 C9 D4	78 STR	ASC	"EDITORE						
92F0:CF D2 C5 AO			DI BLOCCHI						
92F4:CA C9 AO C2			PRODOS"						
92F8:CC CF C3 C3									
92FC:CB C9 AO D0									
9300:D2 CF C4 CF									
9304:D3									
9308:C4 C9 AO CD	79 STR2	ASC	"DI						
930C:C9 D3 D4 C5									
9310:D2 C1 CB CB									
9314:CF AO AD AO									
9318:CF AO AD AO									
931C:CF AO AD AO									
931F:CF AO AD AO									
9323:CF AO AD AO									
9327:CF AO AD AO									
932B:CF AO AD AO									
932F:CF AO AD AO									
9333:CF AO AD AO									
9337:CF AO AD AO									
933B:CF AO AD AO									
933F:CF AO AD AO									
9343:CF AO AD AO									
9347:CF AO AD AO									
934B:CF AO AD AO									
934F:CF AO AD AO									
9353:CF AO AD AO									
9357:CF AO AD AO									
935B:CF AO AD AO									
935F:CF AO AD AO									
9363:CF AO AD AO									
9367:CF AO AD AO									
936B:CF AO AD AO									
936F:CF AO AD AO									
9373:CF AO AD AO									
9377:CF AO AD AO									
937B:CF AO AD AO									
937F:CF AO AD AO									
9383:CF AO AD AO									
9387:CF AO AD AO									
938B:CF AO AD AO									
938F:CF AO AD AO									
9393:CF AO AD AO									
9397:CF AO AD AO									
939B:CF AO AD AO									
939F:CF AO AD AO									
93A3:CF AO AD AO									
93A7:CF AO AD AO									
93AB:CF AO AD AO									
93AF:CF AO AD AO									
93B3:CF AO AD AO									
93B7:CF AO AD AO									
93BB:CF AO AD AO									
93BF:CF AO AD AO									
93C3:CF AO AD AO									
93C7:CF AO AD AO									
93CB:CF AO AD AO									
93CF:CF AO AD AO									
93D3:CF AO AD AO									
93D7:CF AO AD AO									
93DB:CF AO AD AO									
93DF:CF AO AD AO									
93E3:CF AO AD AO									
93E7:CF AO AD AO									
93EB:CF AO AD AO									
93EF:CF AO AD AO									
93F3:CF AO AD AO									
93F7:CF AO AD AO									
93FB:CF AO AD AO									
93FF:CF AO AD AO									

APPLE II

Su un vecchio modello di Apple II che non disponga dei tasti freccia in alto e in basso è possibile ottenere le medesime funzioni premendo Control-J (per la freccia in basso) e Control-K (per la freccia in alto).

Battere il listato

Se disponete dell'assemblatore, battete e assemblate separatamente i listati 1 (Loader) e 2 (l'Editor). Da Basic poi battete Bload Loader.0 Bload Editor.0,A\$2100 Bsave Edit.9200,A\$2000,L2304.

Se non avete l'assemblatore, battete i codici del listato 1, e poi i codici del listato 2, ma a partire dalla locazione \$2100 e non \$9200 come indicato. Salvatelo col Bsave succitato.

Il motivo è semplice: il programma è scritto per girare in alta memoria, ma normalmente quelle locazioni sono riservate a Basic.System, che va "sposta-

to" - ecco lo scopo del Loader - da lì. Il programma viene dunque caricato nella memoria in \$2000 e poi il Loader lo sposta nel punto di residenza quando Basic è stato convinto che quella zona è tabù.

Altre osservazioni

Editor di blocchi è in grado di leggere anche dischi Pascal e Dos 3.3. Per i primi non si presenta alcun problema, poiché sono anch'essi suddivisi internamente in blocchi secondo le medesime regole di ProDos.

I dischi Dos 3.3 sono utilizzati secondo una divisione in settori da 256 byte, ognuno dunque la metà di un blocco ProDos.

Per tradurre il numero di blocco usato da editor di blocchi nel corrispondente numero di settore e traccia dividete il numero di blocco per otto, e otterrete il numero di traccia; dal resto della divi-

sione otterrete il settore consultando la seguente tabella:

Blocco	La 1ª metà equivale a settore	La 2ª metà equivale a settore
0	0	E
1	C	D
2	A	B
3	8	9
4	6	7
5	4	5
6	2	3
7	1	F

Editor di blocchi non è in grado di leggere la maggior parte dei dischi cosiddetti protetti, con alcune notevoli eccezioni; se acquisirete sufficiente comprensione di come lavora il vostro Apple II, sarete in grado di rimuovere la protezione dalla stragrande maggioranza dei dischi protetti (usando l'editor di blocchi come uno degli strumenti necessari, insieme a un copiatore a nibble dotato di editor di traccia).

(Continua)

(La precedente puntata è apparsa sul n. 36)

Seguito listato editor di blocchi

```

9357:F2 E5 A0 E4
935B:E9 A0 C7 AF
935F:DF A0 A0 A0
9363:DE EF EE A0
9367:E5 A7 A0 F5
936B:EE A0 E4 F2
936F:E9 F6 E5 F2
9373:D7 F2 E9 F4
9377:E5 A0 F0 F2
937B:EF F4 E5 E3
937F:FA E5 E4 A0
9383:C2 E9 F4 A0
9387:E1 EC F4 EF
938B:A0 CF AF F3
938F:A0 CF AF F3
9393:CF F0 F4 E9
9397:EF EE E5 A0
939B:E5 F2 F2 E1
939F:F4 E1 A0 A0
93A3:C9 EE E4 E9
93A7:E3 E1 F2 E5
93AB:A0 B1 AC B2
93AF:AE AE AE B7
93B3:C3 EF EE F3
93B7:F5 EC F4 E1
93BB:A0 ED E1 EE
93BF:F5 E1 EC E5
93C3:DE AE A0 E2
93C7:EC EF E3 E3
93CB:EF A0 BE BD
93CF:A0 B0 A1 A0
93D3:C2 F5 E6 E6
93D7:E3 F2 A0 E9
93DB:EC EC E5 E7
93DF:E1 EC E5 A0
93E3:D4 E1 A0 B0
93E7:80 B0 A0 E1
93EB:A0 B1 C6 C6
93EF:A0 A0 A0 A0
93F3:AC 00 92
93F6:
93F6:
93F6:
93F6:8C 6A 97
93FB:A9 D0
93FD:BD 7F 97
9400:A9 03
9402:BD 7F 97
9405:4C 63 97
9408:20 0C FD
940B:A9 00
940D:BD 63 99
9410:BD 61 99
9413:BD 62 99
9416:BD DB BF
9419:20 00 BF
941C:CC
941D:68 99
941F:D0 D5
9421:A9 60
9423:BD 5E 99
9426:
9426:
9426:
9426:
9426:A9 3F
9428:BD 6A BF
942B:A5 74
942D:C9 BA
942F:F0 04
9431:A9 BE
9433:85 74
e' un driver"
"Non
"Write
"Bit
alto posto a"
errata "
"Opzione
"Indicare
1,2...7"
"Consulta
manuale"
"N.
blocco >= 0: "
"Buffer
illegale "
"Da
000 a IFF "
;Dati da POKEare in &
;Resetta htab per MESS
;Resetta uscita per MESS
;Default display hex
;Default blocco numero 0
;Setta LEVEL in GP a zero
;Close all open files
;Default Slot6 Drive1
;Alloca: Chiudi i files aperti
;Abbassa HIMEM delle 8 pagine richieste
;Lasciando IK di buffer per Basic.System
;proteggiti nella Bit Map
;E poi parti!
LDA #3F
STA #BF6A
LDA #74
CMP #8A
BEQ #HIMOK
LDA #8E
STA #74
;Proteggiti in GlobalPage
;HIMEM per caso rivela
;presenza di HELPEDIT?
;se si, non toccarlo!
;Nuovo valore di HIMEM:
;Store it

```

Seguito listato editor di blocchi

```

9A35:A2 00 LDX #0
9A37:BD F3 93 STA AMP$AND,X
9A3A:9D F5 03 LDA $F5,X
9A3D:EB 131 INX #3
9A3E:00 03 CPX #3
9A40:D0 F5 9A37 BNE AND
9A42: 135 ; Richiama la routine che inizializza lo schermo
9A42:20 F2 98 JSR INIT$SCRN
9A45: 137 ; Attende la scelta di una opzione
9A45:A9 00 LDA #0
9A47:85 25 STA #VPOS
9A49:20 22 FC JSR #VTAB
9A4C:8D 7B 05 STA #HTAB
9A4F:20 53 DB JSR #DB$A
9A52:20 53 D5 JSR #INCHR
9A55:85 06 STA #6
9A57:20 96 FC JSR #FC$96
9A5A:A5 06 LDA #6
9A5C:D9 4B JSR #72
9A5E:F0 4D JSR #MSB
9A60:D9 52 JSR #82
9A62:F0 40 JSR #82
9A64:D9 17 JSR #17
9A66:F0 4B JSR #17
9A68:D9 08 JSR #8
9A6A:F0 4B JSR #8
9A6C:D9 15 JSR #15
9A6E:F0 4F JSR #15
9A70:D9 45 JSR #45
9A72:F0 3C JSR #3C
9A74:C9 1B JSR #1B
9A76:F0 4D JSR #4D
9A78:C9 4A JSR #4A
9A7A:F0 2E JSR #2E
9A7C:D9 41 JSR #41
9A7E:F0 71 JSR #71
9A80:C9 46 JSR #46
9A82:F0 4A JSR #4A
9A84:C9 53 JSR #53
9A86:F0 7D JSR #7D
9A88:C9 3F JSR #3F
9A8A:F0 3C JSR #3C
9A8C:C9 42 JSR #42
9A8E:F0 3B JSR #3B
9A90:C9 04 JSR #4
9A92:F0 51 JSR #51
9A94:C9 0B JSR #B
9A96:F0 44 JSR #44
9A98:C9 4D JSR #4D
9A9A:F0 52 JSR #52
9A9C:20 42 99 JSR #42
9A9F:AO 40 JSR #40
9AA1:4C 63 97 JSR #63
9AA4:20 42 97 JSR #42
9AA7:4C 89 97 JSR #89
9AAA:4C 9C 96 JSR #9C
9AB0:4C 98 JSR #98
9AB3:20 80 97 JSR #80
9AB5:4C 85 94 JSR #85
9AB8:20 58 93 JSR #58
9ABD:4C 85 94 JSR #85
9AC2:4C 85 94 JSR #85
9AC5:4C 83 95 JSR #83
9AC8:4C 8C 95 JSR #8C
9ACB:4C 99 95 JSR #99
9ACE:AD 5E 99 JSR #5E
9AD1:49 80 JSR #49
9AD3:BD 5E 99 JSR #BD
9AD6:20 D3 96 JSR #D3
9AD9:4C 85 94 JSR #85
9ADC:20 30 95 JSR #30
9ADF:20 42 97 JSR #42
9AE2:4C 89 97 JSR #89

```

```

204 DEDBLK JSR MINUS
205 JSR READIN
206 SHMBLK JMP MEMBLK
207 BUFFER JMP MEMBUF
208 ASCII LDA #ASCII
209 CMP #00
210 BEQ LETRS
211 LDA #0
212 JMP #XITA
213 LDA #FFF
214 STA #HOMDISP
215 JMP #SHMBLK
216 JSR #CRDUT
217 LDA #31
218 STA #TAB
219 JSR #INCHR
220 CMP #49
221 BMI #BLEAH
222 CMP #56
223 BPL #BLEAH
224 *
225 * Dk, Buana, settiamo il nuovo slot
226 *
227 JSR #80
228 JSR #COUT
229 AND #40F
230 ASL
231 ASL
232 ASL
233 ASL
234 LDX #UNITNUM
235 BMI #LAB1
236 ORA #80
237 STA #UNITNUM
238 JMP #CHOOSE
239 *
240 * Subroutine PLUS
241 * Incrementa NUM_BLOCK di uno e visualizza
242 *
243 INC #BLKNUM
244 BNE #DISPLAY
245 INC #BLKNUM+1
246 JSR #DISPNUM
247 RTS
248 LDA #C061
249 BMI #OPENAPL
250 JSR #BELL
251 LDY #40
252 JSR #MSG
253 OPENAPL LDY #60
254 JSR #MSG
255 BLEAH JSR #BELL
256 LDY #50
257 JSR #MSG
258 MINUS CLD
259 LDY #61
260 JSR #MSG
261 CPY #0
262 BEQ UHM
263 JSR #DISPNUM
264 RTS
265 DEC #BLKNUM+1
266 LDA #BLKNUM+1
267 CMP #FFF
268 BEQ UHM
269 JSR #DISPNUM
270 RTS
271 INC #BLOCK_NUM
272 INC #BLOCK_NUM+1
273 JSR #BELL
274 LDY #70
275 PLA
276 JSR #MSG
277 LDY #0
278 * Effettua l'uscita a BASIC.SYSTEM
279 *
9585:

```

```

94E5:20 56 95 JSR #61
94EB:20 42 97 JSR #42
94EE:4C 89 97 JSR #89
94F1:AD 63 99 JSR #AD
94F4:C9 00 94FD JSR #C9
94F6:F0 05 94FD JSR #F0
94F8:A9 00 94 JSR #A9
94FA:4C FF 94 JSR #4C
94FC:8F FF 94 JSR #8F
9502:4C 89 97 JSR #4C
9505:20 FB DA JSR #20
9508:A9 1F 94 JSR #A9
950A:8D 7B 05 JSR #8D
950C:20 53 D5 JSR #20
950E:C9 31 954E JSR #C9
9510:C9 31 954E JSR #C9
9512:30 3A 954E JSR #30
9514:C9 3B 954E JSR #C9
9516:10 36 954E JSR #10
9518: 224 *
951B: 225 *
951D:09 80 JSR #09
951F:29 0F JSR #29
9521:F0 A JSR #F0
9523:0A 232 JSR #0A
9525:0A 232 JSR #0A
9527:AE 5E 99 JSR #AE
9529:30 02 952A JSR #30
952B:09 80 JSR #09
952D:8D 5E 99 JSR #8D
952F:4C 45 94 JSR #4C
9530: 239 *
9533: 240 *
9535:EE 61 99 JSR #EE
9537:00 03 953B JSR #00
9539:EE 62 99 JSR #EE
953B:20 C5 95 JSR #20
953D:60 247 JSR #60
953F:AD 61 C0 9549 JSR #AD
9541:20 42 99 JSR #20
9543:AO 40 251 JSR #AO
9545:4C 63 97 JSR #4C
9547:4C 63 97 JSR #4C
9549:AO 60 252 JSR #AO
954B:4C 63 97 JSR #4C
954E:20 42 99 JSR #20
9551:AO 50 251 JSR #AO
9553:4C 63 97 JSR #4C
9555:08 248 JSR #08
9557:AE 61 99 JSR #AE
9559:CE 61 99 JSR #CE
955B:EO 00 261 JSR #EO
955F:F0 04 9565 JSR #F0
9561:20 C5 95 JSR #20
9563:60 264 JSR #60
9565:CE 62 99 JSR #CE
9568:AD 62 99 JSR #AD
956B:C9 FF 267 JSR #C9
956D:F0 04 9573 JSR #D
956F:20 C5 95 JSR #20
9571:60 270 JSR #60
9573:EE 61 99 JSR #EE
9575:EE 62 99 JSR #EE
9579:20 42 99 JSR #20
957C:AO 70 274 JSR #C
957E:68 275 JSR #E
9580:4C 63 97 JSR #4C
9583:A2 00 9585 JSR #A2

```

```

;Attualim. hex?
;Si, scegli ASCII.
;No, scegli hex
;... ed esci
;Setta per display ASCII
;Memorizza la scelta
;e visualizza come vuole
;Vtab 2
;Htab 32
;Comanda, padrone!
;Se è minore di uno...
;Mandalo affanculo
;Se è maggiore di sette
;Idem come sopra
;Preparalo al display
;Accetta il n° binario
;Porta n° nei bit 6-5-4
;Drive 1 o 2
;Se flag N=0 (bit 7 = 0)
;dr1, senno dr2
;Sta premendo mela vuota?
;Incrementa il n° blocco.
;E' passato da FF a 00?
;Se si incrementa MSB
;Facciamo settare flag Z)
;Se era 0 (c'è riporto)
;Senno visualizza ed esci
;Dec il MSB di block_nr
;Se era 0, sono cavoli
;Se negativo, ahia!
;Resetta BLOCK_NUM a zero
;Segnala ERR (blocco <07)
;Sporco trucco (POP)

```

APPLE II

Seguito listato editor di blocchi

Address	Instruction	Comment	Address	Instruction	Comment	Address	Instruction	Comment
9585:20 5B FC	JSR	HOME	9618:85 06	354	IND3	9619:81 06	STA	(PAG),Y
9588:A9 15	LDA	#15	961C:85 19	355	IND3	961A:81 06	LDA	\$19
958A:20 00 C3	JSR	EIGHTY	961E:84 1A	356	IND3	961E:84 1A	STY	\$1A
958D:4C D0 03	JMP	\$3D0	9620:A2 00	357	IND3	9620:A2 00	LDX	#0
9590:20 FB DA	JSR	CROUT	9622:BD 4C 93	358	INDRE	9622:BD 4C 93	LDA	STRS,X
9593:A9 0A	LDA	#10	9625:20 ED FD	359	INDRE	9625:20 ED FD	JSR	CDUT
9595:BD 7B 05	STA	HTAB	9628:EB	360	INDRE	9628:EB	INX	#0
9598:20 AC 95	JSR	GETHEX	9629:EO 04	361	INDRE	9629:EO 04	CPX	#4
959B:8D 62 99	STA	BLKNUM+1	962D:AD F5 99	362	INDRE	962D:AD F5 99	BNE	INDRE
959E:20 E3 FD	JSR	PRHEX	9630:10 0A 963C	363	INDRE	9630:10 0A 963C	LDA	HOMDISP
95A1:A0 0B 290	LDA	#11	9632:A5 19	364	INDRE	9632:A5 19	BPL	DAHEX
95A4:BD 61 99	JSR	GETBYTE	9634:20 F9 97	365	INDRE	9634:20 F9 97	LDA	#19
95A6:8D 61 99	STA	BLKNUM	9637:A2 00	366	INDRE	9637:A2 00	JSR	PRINTASC
95A9:4C 45 94	JMP	CHOOSE	9639:4C 43 96	367	INDRE	9639:4C 43 96	LDX	#0
95AC:20 53 D5	JSR	INCHR	963C:A5 19	368	INDRE	963C:A5 19	JMP	IND2
95AF:38	SEC	#48	963E:20 DA FD	369	INDRE	963E:20 DA FD	LDA	\$19
95B0:E9 30	BNE	#48	9641:A2 00	370	INDRE	9641:A2 00	JSR	PRBYTE
95B2:30 FB 95AC	BMI	GETHEX	9643:BD 50 93	371	INDRE	9643:BD 50 93	LDA	#0
95B4:C9 0A	CHP	#10	9646:20 ED FD	372	INDRE	9646:20 ED FD	LDX	STR6,X
95B6:30 OC 95C4	BMI	RETURN	9649:EB	373	INDRE	9649:EB	JSR	CDUT
95B8:3B	SEC	#17	964A:EO 03	374	INDRE	964A:EO 03	INX	#3
95B9:E9 11	SBC	#17	964C:D0 F5 9643	375	INDRE	964C:D0 F5 9643	CPX	#3
95BB:30 EF 95AC	BMI	GETHEX	964E:A0 0F 378	376	INDRE	964E:A0 0F 378	BNE	IND2
95BD:C9 06	CHP	#6	9650:20 9C FC 379	377	INDRE	9650:20 9C FC 379	LDA	#15
95BF:10 EB 95AC	BPL	GETHEX	9653:AD 63 99	378	INDRE	9653:AD 63 99	JSR	\$FC9C
95C1:1B	CLC	#10	9656:10 10 9668	379	INDRE	9656:10 10 9668	LDA	HOMDISP
95C2:69 0A	ADC	#10	9658:20 53 D5	380	INDRE	9658:20 53 D5	EPL	INHEX
95C4:60	RTS	#10	965B:0D 66 99	381	INDRE	965B:0D 66 99	JSR	INCHR
95C5:1	308 * Subroutine DISPLAY_NUM_BLOCK		965E:85 06	382	INDRE	965E:85 06	ORA	HIGHBIT
95C5:1	309 * Subroutine DISPLAY_NUM_BLOCK		9660:20 F9 97	383	INDRE	9660:20 F9 97	STA	#6
95C5:1	310 * Subroutine DISPLAY_NUM_BLOCK		9663:A5 06	384	INDRE	9663:A5 06	JSR	PRINTASC
95C5:A9 01	LDA	#1	9665:4C 6D 96	385	INDRE	9665:4C 6D 96	LDA	#6
95C7:85 25	STA	VPOS	9668:A0 0E	386	INDRE	9668:A0 0E	JMP	DOIT
95C9:20 22 FC	JSR	VTAB	966A:20 1E 97	387	INDRE	966A:20 1E 97	LDA	#15
95CC:A9 0B	LDA	#11	966D:84 1A	388	INDRE	966D:84 1A	JSR	GETBYTE
95CE:8D 7B 05	STA	HTAB	966F:91 06	389	INDRE	966F:91 06	LDA	\$1A
95D1:AD 61 99	LDA	BLKNUM	9671:20 53 D5	390	INDRE	9671:20 53 D5	STA	(PAG),Y
95D4:20 DA FD	JSR	PRBYTE	9674:C9 1B	391	INDRE	9674:C9 1B	JSR	INCHR
95D7:A9 0A	LDA	#10	9676:F0 07 967F	392	INDRE	9676:F0 07 967F	CHP	#27
95D9:8D 7B 05	STA	HTAB	9678:D9 20	393	INDRE	9678:D9 20	BEG	RIFARE
95DC:AD 62 99	LDA	BLKNUM+1	967A:F0 0C 968B	394	INDRE	967A:F0 0C 968B	CMP	#32
95DF:20 E3 FD	JSR	PRHEX	967C:4C 89 97	395	INDRE	967C:4C 89 97	BEP	UNALTRO
95E2:60	322 * Routine EDITOR		967F:A5 19	396	INDRE	967F:A5 19	JMP	SHWBLK
95E3:1	323 * Routine EDITOR		9681:A4 1A	397	INDRE	9681:A4 1A	LDA	\$19
95E3:1	324 * Routine EDITOR		9683:91 06	400	INDRE	9683:91 06	STA	(PAG),Y
95E3:1	325 * Permette di cambiare uno o più bytes del buffer		9685:4C 89 97	401	INDRE	9685:4C 89 97	JMP	SHWBLK
95E3:1	326 * Utilizza: PAG, PAG+1 : indirizzo della pagina		9688:A9 04	402	INDRE	9688:A9 04	LDA	#4
95E3:1	327 * Utilizza: PAG, PAG+1 : valore letto nel byte editato		968A:8D 7B 05	403	INDRE	968A:8D 7B 05	STA	HTAB
95E3:1	328 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		968D:1B	404	INDRE	968D:1B	CLC	#0
95E3:1	329 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		968E:A4 1A	405	INDRE	968E:A4 1A	LDA	\$1A
95E3:1	330 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		9690:CB	406	INDRE	9690:CB	LDY	#1A
95E3:1	331 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		9691:84 1A	407	INDRE	9691:84 1A	INY	#1A
95E3:1	332 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		9693:98	408	INDRE	9693:98	STY	#1A
95E3:1	333 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		9697:20 DA FD	409	INDRE	9697:20 DA FD	TYA	#1A
95E3:1	334 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		969F:14 1A 96	410	INDRE	969F:14 1A 96	JSR	PRBYTE
95E3:1	335 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		9699:4C 1A 96	411	INDRE	9699:4C 1A 96	LDY	\$1A
95E3:1	336 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		969C:A9 00	412	INDRE	969C:A9 00	JMP	IND3
95E3:1	337 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		969E:85 3A	413	INDRE	969E:85 3A	LDA	#0
95E3:1	338 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96A0:AD 60 99	414	INDRE	96A0:AD 60 99	STA	\$3A
95E3:1	339 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96A3:85 3B	415	INDRE	96A3:85 3B	LDA	WHERE+1
95E3:1	340 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96A5:A9 03	416	INDRE	96A5:A9 03	LDA	#3
95E3:1	341 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96A7:85 22	417	INDRE	96A7:85 22	LDA	#3
95E3:1	342 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96A9:20 5B FC	418	INDRE	96A9:20 5B FC	STA	\$22
95E3:1	343 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96AC:20 61 FE	419	INDRE	96AC:20 61 FE	JSR	HOME
95E3:1	344 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96AF:A9 2B 05	420	INDRE	96AF:A9 2B 05	JSR	HOME
95E3:1	345 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96B1:8D 7B 05	421	INDRE	96B1:8D 7B 05	LDA	#40
95E3:1	346 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96B4:20 0C FD	422	INDRE	96B4:20 0C FD	STA	HTAB
95E3:1	347 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96B7:C9 B3	423	INDRE	96B7:C9 B3	JSR	KEYPRS
95E3:1	348 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96B9:FO 0D 96CB	424	INDRE	96B9:FO 0D 96CB	CHP	#83
95E3:1	349 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96BB:AE 60 99	425	INDRE	96BB:AE 60 99	BEG	XITDIS
95E3:1	350 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96BE:EB	426	INDRE	96BE:EB	LDA	WHERE+1
95E3:1	351 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96BF:EB	427	INDRE	96BF:EB	INX	#0
95E3:1	352 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96C0:86 06	428	INDRE	96C0:86 06	STX	#6
95E3:1	353 * Utilizza: PAG, PAG+1 : n° byte editato nella pagina		96C2:A5 3B	429	INDRE	96C2:A5 3B	LDA	\$3B

Seguito listato editor di blocchi

96C4:C5 06	430	CHP	#6	:Siamo in pag. \$WHERE+27	9763:	505 * POINTER e POINT2 servono ad ANI se ProDOS segnali
96C6:D0 E4	431	BNE L		:No, ritorna in loop a	9763:	506 * Errori strani nel caricamento di noi stessi o nel
96C8:20 5B FC	432	JSR HOME			9763:	507 * prepararci la sede di lavoro (righe 100 e dintorni)
96CB:A9 00	433	LDA #0			9763:	
96CD:85 22	434	STA #22		:Resetta la text window	9763:A9 00	LDA #0
96CF:20 14 99	435	JSR INIT2		:Reinizializza screen	9765:85 22	STA VPOS
96D2:4C 45 94	436	JMP CHOOSE			9767:20 22 FC	JSR VTAB
96D5:A2 01	437	DISP	#1	:Posizionati	9768:A9 29	LDA #41
96D7:86 25	438	JMP VPOS			976C:8D 7B 05	STA HTAB
96D9:20 22 FC	439	STX VPOS			976F:A2 00	LDX #0
96DC:A2 16	440	LDA #22			9771:89 53 92	LDA ERR1,Y
96DE:8E 7B 05	441	STX HTAB			9774:20 ED FD	JSR CDUT
96E1:AD 5E 99	442	LDA UNITNUM			9777:EB	INX
96E4:1B	443	CLC			9778:CB	INY
96E5:D8	444	CLD			9779:E0 10	CPX #10
96E6:29 80	445	AND	#80	:Conserva solo n° drive	977B:D0 F4	BNE M
96E8:2A	446	.ROL	#177	:Butta il drive nel Carry	977D:4C 45 94	JMP CHOOSE
96E9:69 B1	447	ADC	CDUT	:Somma ASCII '1'	9780:20 00 BF	MLCAL
96EB:20 ED FD	448	JSR	CDUT	:Visualizzalo!	9783:81	DFB #81
96EE:60	449	RTS		:Esci.	9784:5D 99	DM CMDLIST
96EF:20 FB DA	450	MEMBUF		:VTab 2	9786:D0 C3	BNE ERROR
96F2:A0 2A	451	LDA #42		:Get buffer value	9788:60	RTS
96F4:20 1E 97	452	JSR GETBYTE			9789:AD 63 99	LDA SHMBLK
96F7:C9 C0	453	CHP #C0			978E:C9 00	CMP #0
96F9:F0 0E	454	BED NO			9790:4C 6D 98	BNE SHMASC
96FB:85 06	455	STA #6		:Metti via un attimo...	9793:AD 60 99	JMP SHMASC
96FD:D0 0B	456	AND NO		:E' pari?	9796:85 07	STA WHERE+1
96FF:00 08	457	BNE NO		:No, dunque non va bene	9798:A0 00	LDA #0
9701:A5 06	458	LDA #6		:Si, dunque ricaricalo e	979C:A9 04	STY PAG
9703:8D 60 99	459	STX SHMBLK		:aggiorna pointer a buff	979E:8D 65 99	LDA RIGA
9706:4C 89 97	460	JMP	WHERE+1	: mostra nuovo buff.	97A1:20 C3 97	JSR FOURASC
9709:AF 08	461	LDA #08			97A4:A9 09	LDA #9
970B:BD 60 99	462	STA WHERE+1		:Restore default	97A6:8D 65 99	LDA RIGA
970E:A2 2A	463	LDX #A2			97A9:20 C3 97	JSR FOURASC
9710:BE 7B 05	464	STX HTAB		:Riposizionati	97AC:E6 07	INC PAG+1
9713:20 DA FD	465	JSR PRBYTE		:Visualizza il default	97AE:A0 00	LDA #0
9716:20 42 99	466	JSR BELL			97B0:A9 0E	LDA #14
9719:A0 80	467	LDA #80		:Segnala err e vattene	97B2:8D 65 99	STA RIGA
971B:4C 63 97	468	JMP MESSAGE			97B5:20 C3 97	JSR FOURASC
971E:BC 7B 05	469	STY HTAB		:Get MS Nibble	97B8:A9 13	LDA #19
9721:20 AC 95	470	JSR GETHEX		:Salvalo	97BA:8D 65 99	STA RIGA
9724:85 06	471	STA #6		:Visualizzalo	97BD:20 C3 97	JSR FOURASC
9726:20 E3 FD	472	JSR PRHEX		:Riprendilo	97C0:4C 45 94	JMP CHOOSE
9729:A5 06	473	LDA #6			97C3:A9 00	STA #0
972B:0A	474	ASL			97C5:8D 67 99	LDA INTLOOP
972C:0A	475	ASL			97C8:AE 65 99	SEC QUATAR
972E:0A	476	ASL	#6	:Moltiplicato per 16	97CB:EB	INX
972F:85 06	477	ASL		:Risalvalo	97CC:8E 65 99	STX RIGA
9731:CB	478	INY			97CF:86 25	STX VPOS
9732:8C 7B 05	479	STY HTAB			97D1:20 22 FC	JSR VTAB
9735:20 AC 95	480	JSR GETHEX			97D4:20 1A 9B	JSR RIGASC
9738:85 07	481	STA #7			97D7:AE 67 99	LDX INTLOOP
973A:20 E3 FD	482	JSR PRHEX			97DA:EO 03	CPX #3
973D:A5 07	483	LDA #7			97DC:F0 07	BEQ USCITASC
973F:05 06	484	ORA #6		:Somma il MSNibble	97DE:EB	INX
9741:60	485	AND #6		:End of GET_A_BYTE	97DF:8E 67 99	STX INTLOOP
9742:20 00 BF	486	RTS			97E2:4C CB 97	JMP QUATAR
9745:80	487	JSR READIN			97E5:60	SEC USCITASC
9746:5D 99	488	DFB #80			97E6:38	SBC #80
9748:1D 01	489	DM CMDLIST			97E7:E9 B0	SEC INI
974A:60	490	BNE ERROR			97E9:4C FD 97	JMP INI
974B:60	491	TAY		:Preserva ERRCODE in Y	97ED:69 40	CLC #40
974C:20 42 99	492	ERROR			97EF:84 47	ADC #47
974E:20 27	493	JSR BELL		:Suona l'errore	97F1:20 B0 FE	STY #47
9751:00 05	494	CPY #27		:Se no, vediamo	97F4:A4 47	JSR SETINV
9753:A0 00	495	BNE E1		:Si, va a visualizzare	97F6:4C 0B 98	LDA #47
9755:4C 63 97	496	LDA #0			97F9:	JMP INZ
9758	497	JMP MESSAGE			97F9:	* Routine PRINTASC
975A:4C 63 97	498	CPY #28			97F9:	575 \$ Usata qui sotto e da EDITOR
975B:80 2B	499	BNE E2			97F9:	576 \$ dato un n° hex in A lo visualizza correttamente
975C:A0 05	500	JSR GETA			97F9:	577 \$ Utilizza le soprastanti ADJ1 e ADJ2
975D:A0 10	501	LDA #10			97F9:	
975E:4C 63 97	502	JMP MESSAGE			97F9:	
9761	503	E2			97F9:	
9763:	504	#20		:Deve essere Write prot.	97F9:	
9763:	505	#20		:Visualizza il messaggio d'errore trovato da ERROR	97F9:	

Seguito listato editor di blocchi

```

77F9: C9 B0      580 *      CMP      ##80
77FB: 10 E9      581 PRINTASC BPL      ADJ1
77FD: C9 20      582      CMP      ##20
77FF: 30 EB      583 INI     BMTI     ADJ2
9801: 84 47      584      STY     ##47
9803: 20 84 FE   585      JSR     SETNDRM
9806: 09 80      586      LDA     ##80
9808: 20 ED FD   587 INZ     DRA     ##80
980A: 20 ED FD   588 INZ     JSR     COUT
980C: BA 47      589      STY     ##47
980E: 20 84 FE   590      JSR     SETNDRM
9810: 64 47      591      LDA     ##47
9812: 64 47      592      LDA     ##47
9814: A9 A0      593      JSR     COUT
9816: 20 ED FD   594      JSR     COUT
9819: 60          595      RTS
981A:           596 * End of routine
981A: A2 00      597 RIGRASC LDA     #0
981C: A9 06      598 RIGRASC LDA     #6
981E: B0 7B 05   599      STA     HTAB
9821: B1 06      600 ENTRASC LDA     (PAG),Y
9823: 20 F9 97   601      JSR     PRINTASC
9826: BA          602      TXA
9827: 29 03      603      AND     ##3
9829: C9 03      604      CMP
982B: F0 05      605      BEQ     FOURCAR
982D: EB          606      BEQ     REENTRASC
982E: CB          607      INY
982F: 4C 21 98   608      JMP
9832: E0 0F      609      CPX     FOURCAR
9834: F0 0C      610      BEQ     BLANK
9836: E0 1F      611      CPX     ##1F
983B: F0 10      612      BEQ     NEWLINE
983C: A9 A0      613      LDA     ##A0
983E: 20 ED FD   614      JSR     COUT
983F: 4C 2D 98   615      JMP     REENTRASC
9842: 20 4B F9   616 BLANK  JSR     PR3BLNK
9845: A2 0F      617      LDX     ##F
9847: 4C 2D 98   618      JMP     REENTRASC
984A: CB          619      INY
984B: 60          620      RTS
984C: A9 00      621 BITALD  LDA
984E: B5 25      622      STA     VPOS
9850: 20 22 FC   623      JSR     HTAB
9853: A9 3B      624      LDA     #59
9855: B0 7B 05   625      STA     HTAB
9858: 0D 66 99   626      LDA     HIGHBIT
985B: 49 80      627      EOR     ##80
985D: B0 66 99   628      STA     HIGHBIT
9860: 18          629      CLC
9861: 2A          630      ROL
9862: 2A          631      ROL
9863: A9 80      632      ADC
9865: 20 ED FD   633      JSR     COUT
9868: A0 30      634      LDA     ##30
986A: 4C 63 97   635      JMP     MES5
986D:           636 *
986D: A0 60 99   637 SHIMHEX LDA
9870: B5 07      638      STA     PAG+1
9872: 60 00      639      LDY     #0
9874: 84 06      640      STY     #4
9876: B0 65 99   641      LDA     #80
987B: 20 9D 98   642      STA     RIGA
987E: A9 09      643      JSR     FOURM
9880: B0 65 99   644      LDA     #9
9883: 20 9D 98   645      STA     RIGA
9886: E6 07      646      JSR     FOURM
9888: A0 00      647      INC     PAG+1
988A: A9 0E      648      LDY     #0
988C: B0 65 99   649      LDA     #14
988F: 20 9D 98   650      STA     RIGA
9892: A9 13      651      JSR     FOURM
9894: B0 65 99   652      LDA     #19
9897: 20 9D 98   653      STA     RIGA
989A: 4C 45 94   654      JMP     CHOOSE
989D:           655 *
989D:           656 * Subroutine FOURM
989D:           657 * Stampa quattro righe consecutive di numeri hex
989D:           658 * Presi dalla PAG settata alla posizione Y
989D:           659 * Chiama UNARIGA quattro volte
989D:           660 * In entrata, RIGA è il VTAB della 1^ riga MEND UNO
989D:           661 * In uscita, RIGA è incrementato di 4
989D:           662 *
989D:           663 *
989D: A9 00      664 FOURM  LDA     #0
989F: BD 67 99   665      STA     INTLOOP
98A2: AE 65 99   666      LDX     RIGA
98A5: EB          667      INX
98A8: BE 65 99   668      STX     RIGA
98AB: B6 2E      669      STX     VPOS
98AD: B6 2E      670      JSR     VTAB
98AE: 20 22 FC   671      LDA     UNARIGA
98B0: 20 C0 98   672      LDX     INTLOOP
98B1: AE 67 99   673      CPX     #3
98B4: F0 07      674      BEQ     USCITA
98B6: F0 07      675      BEQ     USCITA
98B9: BE 67 99   676      STX     INTLOOP
98BB: BE 67 99   677      JMP     FOUR
98BC: 4C A2 98   678 USCITA JMP     FOUR
98BF:           679 *
98C0:           680 * Subroutine UNARIGA
98C0:           681 * Stampa una riga di numeri hex, all'attuale vtab
98C0:           682 * Y contiene in entrata il 1^ n^ di byte da stampare
98C0:           683 * (da 0 a 1E0)
98C0:           684 * Pag deve contenere $XPO0 oppure $XD00, P pari D disp.
98C0:           685 *
98C0: A2 00      686 UNARIGA LDA     #0
98C2: A9 04      687      STA     HTAB
98C4: BD 7B 05   688      LDA     (PAG),Y
98C7: B1 06      689      JSR     PRBYTE
98C9: 20 DA FD   690      TXA
98CC: B8          691      AND     ##3
98CD: 29 03      692      CMP     ##3
98CF: C9 03      693      BEQ     ALTR14
98D1: F0 05      694      BEQ     REENTRY
98D3: EB          695      INY
98D4: CB          696      INY
98D5: 4C C7 98   697      JMP     ENTRY
98D8: E0 0F      698 ALTR14 CPX     ##F
98DA: F0 0C      699      BEQ     SPAZI
98DC: E0 1F      700      CPX     ##1F
98DE: F0 10      701      BEQ     ALTRARIGA
98E0: A9 A0      702      LDA     ##A0
98E2: 20 ED FD   703      JSR     COUT
98E3: 4C D3 98   704      JMP     REENTRY
98E6: 20 4B F9   705      STA     SPAZI
98E8: A2 0F      706      LDA     ##F
98EB: 4C D3 98   707      JMP     REENTRY
98F0: CB          708 ALTRARIGA INY
98F1: 60          709      RTS
98F2:           710 *
98F2:           711 * Fine della visualizzazione hex (SHOW_BLOCK_HEX)
98F2:           712 * e routine connesse
98F2:           713 INITSCRN JSR     EIGHTY
98F5: 20 00 C3   714      JSR     CROUT
98F8: A2 00      715      LDX     #0
98FA: BD 51 92   716 LOOP  LDA     MSG1,X
98FD: EB          717      INX
98FE: 20 ED FD   718      JSR     COUT
9901: E0 4F      719      CPX     #79
9903: D0 F5      720      BNE     LOOP
9905: 20 FB DA   721      JSR     CROUT
9908: A2 00      722      LDX     #0
990A: A9 DF      723      LDA     #223
990C: B0 65 99   724 LOOP2 STA     RIGA
990E: 20 ED FD   725      JSR     COUT
9910: E0 4F      726      CPX     #79

```

Seguito listato editor di blocchi

```

9912:00 F8 990C 727 728 INITZ          BNE LOOP2
9914:A9 00          LDA HTAB
9916:8D 78 05          STA HTAB
9919:A9 05          LDA #5
991B:85 25          STA VPOS
991D:20 22 FC          JSR VTAB
9920:          INITSREEN, stampa gli indirizzi del buffer
9920:A2 00          LDX #0
9922:A0 00          LDA MS63,X
9924:8D A0 92          STA COUT
9927:20 ED FD          JSR COUT
992A:EB          INX
992B:EO 4C          CPX #76
992D:FO 08          BEQ VIDEO
992F:CB          INY
9930:CO 04          BEQ CR
9932:FO 06          BEQ CR
9934:4C 24 99          JMP LUP
9937:4C 89 97          JMP SHMBLK
993A:20 FB DA          JSR CROUT
993D:A0 00          LDX #0
993F:4C 24 99          JMP LUP
9942:A9 20          LDA #20
9944:8D 64 99          STA LENGTH
9947:A9 02          LDA #2
9949:20 AB FC          JSR WAIT
994C:8D 30 C0          STA SPKR
994F:A9 24          LDA #24
9951:20 AB FC          JSR WAIT
9954:8D 30 C0          STA SPKR
9957:CE 64 99          DEC LENGTH
995A:DE EB          BNE BELL1
995C:60          RTS
995D:03          DFB #03
995E:60          DFB #60
995F:00 08          DFB #00,$08
9961:00 00          DFB #00,$00
9963:00          DFB #0
9964:00          DFB #0
9965:          DFB #0
9966:00          DFB #0
9967:          DFB #0
9968:4B AD 30          DFB #4B,$AD,$30
996B:01          DFB #01
996C:00          DFB #0

```

;Param_count per
 ; READ_BLOCK
 ;Buffer per il blocco

 ;ProbELL

 ;Inizio di Basic.System
 ;Param_count per Close
 ;Close all files

```

** SUCCESSFUL ASSEMBLY := NO ERRORS
** ASSEMBLER CREATED ON 15-JAN-84 21:28
** TOTAL LINES ASSEMBLED 772
** FREE SPACE PAGE COUNT 78

```

Arrivederci su Bit!

Sì, **Personal Software** si trasferisce all'interno di *Bit*, la prima e più diffusa e tecnicamente all'avanguardia rivista di personal computer e accessori, mensile anch'esso edito dal Gruppo Editoriale Jackson.

Quindi, dal prossimo mese, cercate in edicola *Bit*, famosa per dinamicità e, soprattutto, capacità professionale di trattare ogni aspetto dell'informatica a noi più prossima, quella dei pc.

Sarete così meglio aggiornati sul panorama dei personal computer, pur avendovi riservato uno spazio tutto dedicato ai programmi. Su *Bit* avrete la possibilità di un aggiornamento globale più completo, potrete essere coinvolti in avventure come la telematica e le comunicazioni via modem, potrete imparare come scrivere i vostri programmi e quali pacchetti sono già pronti.

Sarete tempestivamente avvertiti di importanti avvenimenti che riguardino software e hardware, come il prossimo Call for Game a cui siete tutti invitati a partecipare.

Ed è a tutti coloro i quali ci hanno seguiti fino a questo momento che non diciamo addio, ma un vero arrivederci al prossimo mese tutti con *Bit*. Lo troverete anch'esso in edicola o vi verrà spedito in abbonamento.

Grafica a tre dimensioni

Un'introduzione al Cad

di Valerio Anselmo

Come si sa, la sigla Cad indica "Computer aided design", cioè disegno con l'ausilio dell'elaboratore. Il sofisticato gruppo di istruzioni grafiche del C 16 permette di emulare, naturalmente in ambito limitato, ciò che è possibile ottenere in questo campo con computer dedicati, dotati di velocità, capacità grafiche e di memoria decisamente superiori.

Non si pretende di fare l'impossibile con il C 16. Si sono rese necessarie scelte limitative perché programma e dati rientrassero nei 2 Kbyte (o poco più) disponibili in modo grafico. Per esempio, si è dovuto rinunciare a una grafica multicolore, sia perché le lettere del testo sarebbero diventate pressoché illeggibili, sia perché le richieste relative ai colori avrebbero tolto troppo spazio ai dati. Si è dovuto rinunciare ad avere più di una ventina di dati (segmenti, ellissi) per non togliere troppo spazio al programma. Insomma, ciò che vedete pubblicato è piuttosto un compromesso fra quanto si sarebbe potuto fare e quanto la memoria a disposizione permetteva in realtà di fare. Il risultato comunque è interessante e utile, soprattutto come introduzione al problema della rappresentazione di oggetti nello spazio e alla possibilità di vederli da angolazioni diverse dopo la memorizzazione dei dati.

La rappresentazione di un oggetto nello spazio si può ottenere fondamentalmente in due modi, entrambi basati sulla proiezione dell'oggetto stesso su un piano. Il primo metodo è impropriamente chiamato di rappresentazione "tridimensionale", in quanto l'oggetto, anche se si può presentare sotto angolazioni e prospettive diverse, è in realtà riprodotto su un piano e visto con visione non stereoscopica. È il tipo di rappresentazione ottenibile con questo programma. L'altro tipo di rappresentazione fornisce una visione realmente stereoscopica, ma richiede l'uso di occhiali speciali (di solito con lenti polarizzanti) e di schermi polarizzanti. L'immagine sarà in realtà formata da due immagini, una per l'occhio destro e una per l'occhio sinistro. Si stanno facendo ora anche esperimenti di visione stereoscopica a parallasse verticale, che non richiede l'uso degli occhiali di cui sopra, ma siamo ancora allo stadio di studio.

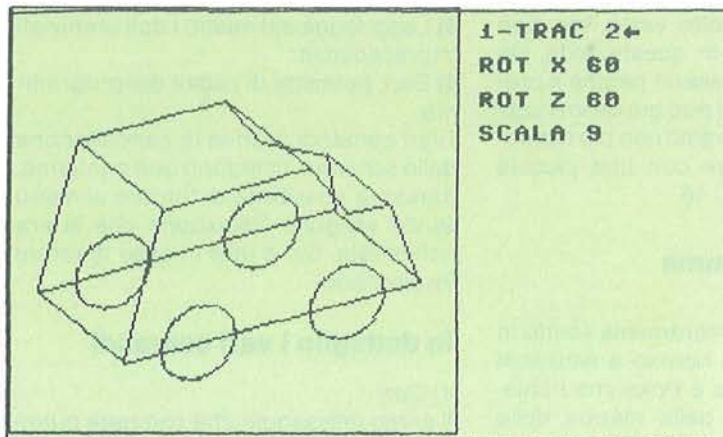


Figura 1 - Esempio di tracciamento di figura ruotata di 60°.

Quindi quella che di solito viene presentata come una rappresentazione tridimensionale degli oggetti in realtà non è altro che la rappresentazione bidimensionale di un oggetto, con la possibilità però di variare il punto di vista visualizzando una proiezione "spazia-

le" dell'oggetto memorizzato. Esempi a tutti noti sono quelli del disegno meccanico, dove un motore o un pezzo da far preparare da un "machining center" viene disegnato in modo tale che possa essere ruotato, ingrandito o evidenziato in una delle sue parti sullo schermo,

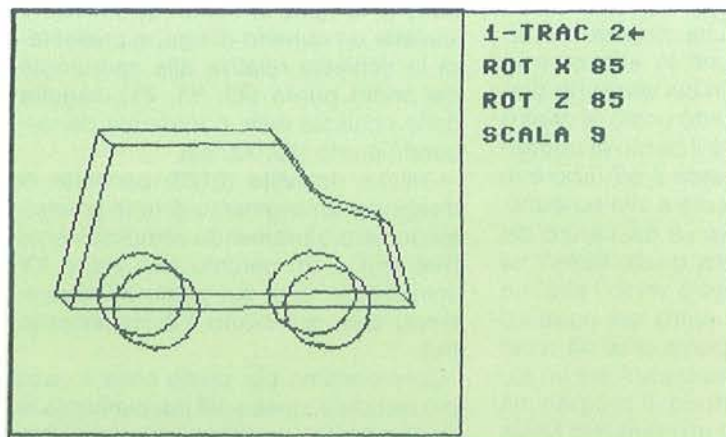


Figura 2 - Esempio di tracciamento di figura ruotata di 85°.

così da accertarne l'esattezza prima dell'esecuzione vera e propria del prototipo. È in via di soluzione definita il passaggio poi dal Cad al Cam, cioè dal disegno così tracciato all'esecuzione del pezzo mediante macchine fresatrici automatiche, anch'esse guidate da computer.

Nell'ambito della progettazione edilizia, per esempio, anziché utilizzare dei modellini plastici è possibile lasciare all'elaboratore il compito di rappresentare sullo schermo, a colori e con notevole realismo, il risultato finale del progetto stesso. Avendo inoltre la possibilità di variarne a piacimento dimensioni, colori e altre caratteristiche.

L'argomento è molto vasto, ma non sarà approfondito in questa sede. Un cenno è stato necessario perché il programma presentato può già far immaginare quanto, anche solo con più memoria, si potrebbe fare con una piccola macchina come il C 16.

Uso del programma

Il programma è interamente scritto in Basic, senza alcun ricorso a istruzioni speciali, come Peek e Poke che richiedono conoscenze della mappa della memoria, che esulano dal Basic vero e proprio. Il programma è poi piuttosto breve (per forza di cose), tanto da poter essere immesso in macchina in poco tempo. Dopo aver battuto il testo del listato e averlo registrato su nastro, premete contemporaneamente i tasti [Shift] e [F 3] per far avviare il computer. Si presenta una prima videata con il menu principale. Come si noterà, il testo è ridotto all'osso, sempre per poter strappare qualche byte al programma e lasciarlo ai dati. Sulla sinistra c'è un riquadro di 200 punti in altezza e in larghezza. È l'area in cui verranno presentati i grafici. Il punto posto al centro di questo quadrato è il punto di coordinate $X=0$, $Y=0$. L'asse Z all'inizio è in posizione perpendicolare allo schermo. Il semiasse X positivo va dal centro del riquadro verso destra, quello delle Y va dal centro del riquadro verso l'alto. La distanza utile dal centro del riquadro verso i lati è di 99 punti, cifra da tener presente per poter rappresentare un solido in modo opportuno. Il programma contiene comunque un comando Scala

per l'ingrandimento o il rimpicciolimento dell'immagine, per ovviare a problemi legati alle dimensioni improprie dell'originale.

I sei comandi che si possono scegliere dal menu principale sono:

- 1) **Dati**: permette l'introduzione delle coordinate dei segmenti di retta o delle ellissi (o circonferenze);
- 2) **Trac**: permette il tracciamento del disegno con la scelta del punto di vista (rotazione sull'asse X, e/o sull'asse Z) e delle dimensioni;
- 3) **Canc**: cancella lo schermo riportandolo alle condizioni iniziali;
- 4) **Arch**: archivia su nastro i dati del disegno;
- 5) **Legg**: legge dal nastro i dati archiviati in precedenza;
- 6) **Esci**: permette di uscire dal programma.

I vari comandi, tranne la cancellazione dello schermo, chiedono una conferma, dando la possibilità di tornare al menu senza eseguire l'istruzione che si era richiamata. Ciò è utile in caso di errore involontario.

In dettaglio i vari comandi

1) Dati

Il primo messaggio che compare quando si preme il tasto [1] è 1-Dati 2←. Ciò significa che, se abbiamo premuto il tasto per errore e vogliamo tornare al menu, dovremo premere [2]. Se premiamo [1] appare la parola Linea, che richiede il numero progressivo della linea (segmento, cerchio o ellisse) da tracciare. Se si preme il tasto [Return] senza battere alcun numero, il valore sarà considerato come uguale a zero e si tornerà alla richiesta iniziale. Ripremendo [Return] si tornerà al menu. Se invece si immette un numero di riga, si presenterà la richiesta relativa alle coordinate del primo punto (X_1 , Y_1 , Z_1), seguita dalla richiesta delle coordinate del secondo punto (X_2 , Y_2 , Z_2).

L'ultima richiesta (0123) permette di disegnare un segmento di retta (immettendo zero o premendo semplicemente [Return]) o un cerchio (sul piano XY immettendo uno, sul piano XZ immettendo due, sul piano YZ immettendo tre).

Consideriamo per prima cosa il caso più semplice, quello del tracciamento di un segmento. Vogliamo immettere i dati

per il tracciamento di un cubo. Le coordinate che dobbiamo immettere sono quelle dei vari segmenti che costituiscono il solido in questione, in pratica le coordinate dei vertici del cubo. Da quanto detto in precedenza, sceglieremo di rappresentare il cubo centrato sullo schermo (per una rotazione più regolare) e terremo i valori tra -50 e +50 sui tre assi (X, Y e Z). Potremo così fare esperimenti sia di rotazione degli assi che di rappresentazione in scala (ingrandimento o rimpicciolimento) dell'oggetto. Il programma permette di tracciare fino a 21 linee. Questo è un limite che purtroppo si fa sentire in modo pesante.

Per rappresentare un cubo, comunque bastano 12 segmenti, con le seguenti coordinate:

Segmento	X1	Y1	Z1	X2	Y2	Z2
N. 1	-50	50	50	50	50	50
N. 2	50	50	50	50	-50	50
N. 3	50	-50	50	-50	-50	50
N. 4	-50	-50	50	-50	50	50
N. 5	-50	50	50	-50	50	-50
N. 6	50	50	50	50	50	-50
N. 7	50	-50	50	50	-50	-50
N. 8	-50	-50	50	-50	-50	-50
N. 9	-50	50	-50	50	50	-50
N. 10	50	50	-50	50	-50	-50
N. 11	50	-50	-50	-50	-50	-50
N. 12	-50	-50	-50	-50	50	-50

Quando si immettono i dati, si può correggere cancellando con il tasto Del e ribattendo. Il cursore invece non si vedrà perché siamo in modo grafico. Per immettere il numero premere, come al solito, il tasto [Return]. Anche qui, la pressione di [Return] senza che si sia battuto alcun numero, significa l'immissione di uno zero in quella posizione. All'ultima domanda (0123) basterà battere [Return] per indicare che si vuole un segmento e non un cerchio (oppure immettere uno zero).

Dopo aver immesso i dati di una linea, il programma ritorna al menu. Da questo, si deve ancora premere il tasto [1] per tornare all'immissione dei dati, fino a che non sia stato fatto per tutti i dati. Se per caso ci si accorge di aver immesso dei dati errati, niente paura: basta reinserire i dati giusti per quella linea e tutto andrà a posto.

2) Trac e 3) Canc

Una volta immessi tutti i dati, possiamo

COMMODORE 16

vedere il nostro cubo. Premiamo, dal menu, il tasto [2] (Trac) e poi il tasto [1]. Premendo poi semplicemente il tasto [Return] alle varie richieste che si presentano (Rot X, Rot Z, Scala), otterremo la rappresentazione dell'oggetto visto dall'alto. Visto così il cubo naturalmente ci appare come un quadrato.

Ripuliamo lo schermo premendo il tasto [3] (Canc) e poi ripremiamo il tasto [2] (Trac). Questa volta proveremo a vedere il cubo sotto un altro profilo. Faremo prima ruotare l'asse X di 60 gradi (il cubo ruoterà in modo che la faccia superiore si allontani da noi verso l'esterno), poi l'asse Z (che ora si troverà quasi in posizione verticale) di 30 gradi. La figura del cubo che si presenta ai nostri occhi è ora molto più realistica. La prospettiva è ortogonale. Ripuliamo ancora una volta lo schermo e facciamo ridisegnare il nostro cubo da un altro punto di vista. Facciamolo ruotare di -15 gradi sia sull'asse X che sull'asse Z, e diamogli un rapporto di scala di -3. Il cubo che si presenta ora sullo scher-

mo è stato ruotato leggermente verso di noi e sulla sinistra, ed è stato ridotto. Facciamo altri esperimenti con la rappresentazione del cubo, aumentandone le dimensioni e variando la rotazione degli assi. Gli assi possono essere ruotati sia in senso positivo che in senso negativo, e le rotazioni sono in gradi. Per quanto riguarda le dimensioni, si consiglia di non esagerare troppo: i valori suggeriti per la scala vanno da -9 a +9. Nel caso di questo solido, quando si chiede l'ingrandimento 9 il disegno sconfinava fuori dal quadrato, ma le linee fuori campo vengono poi cancellate alla presentazione del menu. Una limitazione del tracciamento entro i confini dell'area a disposizione avrebbe allungato irrimediabilmente il programma togliendo ancor più spazio ai dati.

Avrete notato che i dati immessi sono tutti numeri interi. Anche questa è un'altra delle limitazioni imposte dalla poca memoria a disposizione. Se si immettono numeri frazionari, questi saranno arrotondati all'intero più vicino.

4) Arch

Se si vuole, si può ora archiviare il disegno su nastro. Mettere nel registratore un nastro per i dati, riavvolgerlo e, mentre sullo schermo c'è il menu, premere il tasto [4]. Premere poi ancora [1] e, quando compare la parola Record, premere il tasto Record sul registratore. I dati del disegno vengono memorizzati su nastro. Quando l'archiviazione del disegno è completata, ricompare sullo schermo il menu iniziale.

5) Legg e 6) Esci

Per prova, riavvolgiamo il nastro e vediamo se i dati sono stati registrati come si deve. Dal menu iniziale premere il tasto [6] (Esci) e poi [1] (Fine). Riavviare poi il programma con [Shift] [F 3] e, alla comparsa del menu, premere il tasto [2] (Trac) e far tracciare il disegno per accertarci che in memoria non ci siano più dati. In mezzo al riquadro del disegno comparirà un puntino. Quando ricompare il menu, premere il tasto [5] (Legg) e poi ancora [1]. Alla scritta Play preme-

Listato 1 - Il programma Grafica a tre dimensioni.

```

0 M=20:DIMEZ(M,6)
1 GRAPHIC1,1:BOX,0,0,199,199
2 GOSUB30:RESTORE3:FORJ=0TO5:READK$:CHAR
,26,J*2,K$:NEXTJ
3 DATA1-DATI,2-TRAC,3-CANC,4-ARCH,5-LEGG
,6-ESCI
4 GETKEYK$:IFK$<"1"OR"K$>"6"THEN4
5 GOSUB30:ONVAL(K$)GOTO7,17,1,31,34,6
6 CHAR,26,0,"1-FINE 2+":GETKEYK$:IFK$<>"
1"THEN2
7 CHAR,26,0,"1-DATI 2+":GETKEYK$:IFK$<>"
1"THEN2
8 T$="LINEA":R=2:GOSUB11:I=0-1:IFI<0ORI>
MTHEN7:ELSEIFX<ITHENX=I
9 RESTORE10:FORJ=0TO6:READT$:R=J*2+4:GOS
UB11:E%(I,J)=0:NEXTJ:GOTO2
10 DATA1,Y1,Z1,X2,Y2,Z2,0123
11 R$="":CHAR,26,R,T$:S=LEN(T$)
12 GETKEYK$:IFK$=CHR$(13)THENQ=VAL(R$):R
ETURN
13 W=LEN(R$):IFK$<>CHR$(20)THEN15
14 IFR$<>" "THENR$=LEFT$(R$,W-1):K$=" " :G
OTO16:ELSE12
15 IFS+W<13THENR$=R$+K$:W=W+1:ELSE12
16 CHAR,26+S+W,R,K$:GOTO12
17 CHAR,26,0,"1-TRAC 2+":GETKEYK$:IFK$<>"
1"THEN2
18 RESTORE19:FORJ=1TO3:READT$:R=J*2:GOSU

```

```

B11:F(J-1)=Q:NEXTJ:F(2)=1+F(2)/10
19 DATAROT X,ROT Z,SCALA
20 G=PI/180:O=COS(F(1)*G):P=SIN(F(1)*G):L
=COS(F(0)*G):N=SIN(F(0)*G):FORI=0TOX
21 H=E%(I,0):K=-E%(I,1):O=E%(I,2):R=E%(I
,3):S=-E%(I,4):T=E%(I,5):V=E%(I,6)
22 IFV<1ORV>3THEN26:ELSEW=R-H:Y=K-S:Z=Q-
T:MX=R-W:MY=K-Y:MZ=Q-Z
23 FORJ=12TO360STEP12:U=PI*(J+90)/180
24 IFV=1THENR=MX+COS(U)*W:S=MY+SIN(U)*Y:
GOTO26
25 IFV=2THENR=MX+COS(U)*W:T=MZ+SIN(U)*Z:
ELSES=MY+Y+COS(U)*Y:T=MZ+SIN(U)*Z
26 A=H*O-K*P:B=R*O-S*P:C=K*O+H*P:G=S*O+R
*P:C=C*L-Q*N:G=G*L-T*N
27 A=A*F(2):C=C*F(2):B=B*F(2):G=G*F(2)
28 DRAW,A+99,C+99TO+(B-A),+(G-C):IFV<>0T
HENH=R:K=S:Q=T:NEXTJ
29 NEXTI:GOTO2
30 FORK=0TO24:CHAR,25,K,"
":NEXTK:RETURN
31 CHAR,26,0,"1-ARCH 2+":GETKEYK$:IFK$<>"
1"THEN2:ELSECHAR,26,2,"RECORD"
32 OPEN1,1,1,"D":PRINT#1,X:FORI=0TOX:FOR
J=0TO6:PRINT#1,E%(I,J)
33 NEXTJ,I:CLOSE1:GOTO2
34 CHAR,26,0,"1-LEGG 2+":GETKEYK$:IFK$<>"
1"THEN2:ELSECHAR,26,2,"PLAY"
35 OPEN1,1,0,"D":INPUT#1,X:FORI=0TOX:FOR
J=0TO6:INPUT#1,E%(I,J)
36 GOTO33

```

COMMODORE 16

re il tasto Play sul registratore. Il C 16 comincerà a leggere i dati e, quando li avrà letti tutti, ripresenterà il menu iniziale. Ora i dati del disegno sono di nuovo in macchina e possono essere richiamati con il tasto [2] (Trac).

Il tracciamento delle curve

Abbiamo così visto in breve tutti i comandi del menu principale. Resta da chiarire solo il tracciamento delle curve (cerchi o ellissi).

Il programma permette di tracciare dei cerchi dando due soli punti e stabilendo il piano su cui queste curve devono essere tracciate. Il programma si aspetta che vengano date per prime le coordinate del punto posto in alto al centro (nel caso del cubo visto prima, per esempio, nel piano XY saranno le coordinate $X1=0$, $Y1=50$ e $Z1=50$) e poi le coordinate del punto posto a 90 gradi in senso orario (nell'esempio proposto saranno le coordinate $X2=50$, $Y2=0$ e $Z2=50$). Il numero da immettere alla richiesta 0123 sarà 1 se, come nell'esempio proposto, la curva si trova nel piano XY, sarà 2 se la curva si trova nel piano XZ e sarà 3 se si trova nel piano YZ. Se il valore della terza coordinata (quella perpendicolare al piano) sarà diverso, verrà assunto come valido il valore del secondo punto (nel caso presentato, il valore di Z2). Se i valori non sono dati nell'ordine stabilito, potrà essere visualizzato un peduncolo tra la posizione iniziale della curva e la posizione iniziale del tracciato vero e proprio e, nella peggiore delle ipotesi, verrà solo tracciato un segmento o un qualcosa di imprevisto. Le curve perciò vanno studiate bene prima di immettere i dati. Per esempio, per far tracciare una figura formata da 6 ellissi e 3 cerchi che si intersecano al centro, dovremo introdurre i seguenti dati:

Linea	X1	Y1	Z1	X2	Y2	Z2	0123
N. 1	0	60	0	30	0	0	1
N. 2	0	30	0	60	0	0	1
N. 3	0	0	60	30	0	0	2
N. 4	0	0	30	60	0	0	2
N. 5	0	0	60	0	30	0	3
N. 6	0	0	30	0	60	0	3
N. 7	0	60	0	60	0	0	1
N. 8	0	0	60	60	0	0	2
N. 9	0	0	60	0	60	0	3

Provate ora a far tracciare la figura, lasciando a zero tutti i valori (Rot X, Rot Z, Scala). La figura che ne risulta non è molto complessa perché è vista dall'alto. Ma basta inclinare un po' questo oggetto sugli assi e ne verrà fuori tutta la complessità. L'esercizio è utile per capire esattamente come avviene il tracciamento delle ellissi.

In possesso di questi dati, possiamo dedicarci ora all'esecuzione di un disegno rappresentante in forma molto semplificata un'automobile. La limitazione del numero di linee a disposizione ci costringe a farne poco più di un abbozzo, che comunque darà un'idea della forma dell'oggetto ruotato nello spazio. Le coordinate dei 21 punti sono le seguenti:

Linea	X1	Y1	Z1	X2	Y2	Z2	0123
N. 1	-20	40	-20	20	40	-20	0
N. 2	20	40	-20	-20	-40	-20	0
N. 3	20	-40	-20	-20	-40	-20	0
N. 4	-20	-40	-20	20	-40	-20	0
N. 5	-20	40	-20	15	35	0	0
N. 6	-15	35	0	15	35	0	0
N. 7	15	35	0	40	-20	0	0
N. 8	-15	35	0	-20	20	5	0
N. 9	-20	20	5	-20	10	20	0
N. 10	-20	10	20	20	10	20	0
N. 11	20	10	20	20	20	5	0
N. 12	20	20	5	15	35	0	0
N. 13	-20	10	20	-20	-30	20	0
N. 14	-20	-30	20	20	-30	20	0
N. 15	20	-30	20	20	10	20	0
N. 16	-20	-30	20	-20	-40	-20	0
N. 17	20	-30	20	20	-40	-20	0
N. 18	-20	20	-10	-20	-30	-20	3
N. 19	-20	-20	-10	-20	-10	-20	3
N. 20	20	20	-10	20	30	-20	3
N. 21	20	-20	-10	20	-10	-20	3

Una volta immessi i dati, registriamoli su nastro. Passiamo poi alla visualizzazione del disegno. Siccome le misure sono piuttosto piccole rispetto al normale, potremo ora usare il comando Scala per ingrandirne le dimensioni. Può andar bene anche un fattore di 9. Quando viene fatto tracciare senza cambiare i valori predefiniti (rotazione di 0 gradi sia per l'asse X che per l'asse Z), vedremo l'automobile dall'alto. Per vederla in posizione di marcia, cancelliamo lo schermo e facciamo tracciare il disegno con gli assi X e Z ruotati entrambi di 90 gradi. Ecco la macchina in assetto di marcia. Una rotazione di 60 gradi su entrambi gli assi ce la presenta dall'alto e di fianco, mentre una rotazione di 90 gradi del solo asse X ce la presenterà vista dalla parte posteriore.

Questo è già un buon esempio di quanto si può ottenere con un programma per il disegno assistito dall'elaboratore.

Uno sguardo al listato

Per semplicità, verranno indicati i numeri di riga e accanto a essi i relativi commenti.

0 - La numerazione delle righe parte da 0 e prosegue per incrementi di 1. Ciò permette di risparmiare memoria. La variabile M definisce il numero di linee che il programma potrà tracciare.

1 - L'istruzione grafica Box traccia il riquadro in cui verranno riprodotte le proiezioni dei solidi.

2-5 - Presentazione del menu iniziale.

6 - Uscita dal programma previa conferma.

7-10 - Immissione dei dati. La variabile X memorizza il numero massimo di linee utilizzato. Ciò servirà per rendere in seguito più rapida l'archiviazione e la lettura dei dati sul nastro.

11-16 - Subroutine di analisi dei dati immessi e di presentazione delle scritte sullo schermo grafico. Riconosce la pressione del tasto [Return] come valida per l'immissione dei dati, oltre a permettere la cancellazione con il tasto Del (si vedano le righe 13 e 14).

17-19 - Richiesta dei dati per il tracciamento. Da notare, alla riga 18, il meccanismo per la definizione della Scala.

20-29 - Routine di tracciamento. Le righe 22-25 servono a determinare i punti delle circonferenze (o ellissi) per il calcolo. L'incremento scelto è di 12 gradi. Se non lo si ritenesse soddisfacente come accuratezza, lo si potrà variare nella riga 23. Si tenga però presente che, a causa dei molti calcoli e dei vari arrotondamenti conseguenti, le circonferenze non saranno mai perfette come se fossero tracciate con l'istruzione Circle.

30 - Subroutine di cancellazione della parte destra dello schermo.

31-33 - Subroutine di archiviazione dei dati.

34-36 - Subroutine di lettura dei dati da nastro. ■

Basato su un noto gioco arcade, questo programma vede il vostro piccolo personaggio costretto a salire e scendere precipitosamente le scale che collegano i diversi livelli, inseguito dalle tremende creature verdi. Per sopravvivere esso dovrà distruggerle, facendole precipitare nelle buche da lui scavate, per poi seppellirle senza pietà.

Purtroppo, gli elementi in suo favore non sono molti; la riserva di ossigeno è infatti limitata, ed eccessivi movimenti o un tempo troppo lungo speso nella caccia possono significare la sua fine. Ovviamente anche la cattura gli sarebbe letale.

Tutto, dalla grafica, al colore, al suono, contribuisce a rendere appassionante questo programma, che vi terrà per ore inchiodati alla sedia. State calmi e non fatevi prendere dal panico!

Il programma

Copiate il programma 1, per creare i caratteri grafici e fatelo girare. Quindi

Listato 1 - Il primo programma Panico!

```
10 FOR i=USR "a" TO USR "a"+145
20 READ x
30 POKE i,x
40 NEXT i
50 DATA 0,24,24,255,189,60,36,102
60 DATA 195,66,126,90,126,126,36,66
70 DATA 255,255,24,24,24,24,24,255
80 DATA 195,195,195,255,255,195,195,195
90 DATA 0,28,28,12,60,12,22,50
100 DATA 0,56,56,48,60,48,104,76
110 DATA 12,16,40,68,66,1,0,0
```

Listato 2 - Il secondo programma Panico!

```
10 POKE 23693,56: CLS : GO TO 575
20 CLS : DEF FN R(X)=INT (RND*X):
GO TO 610
30 LET X1=X: LET Y1=Y
40 IF INKEY$="H" THEN GO SUB 750
50 IF X=3 OR X=8 OR X=13 OR X=18
THEN LET Y=Y+(IN 61438=251
AND Y<31)-(IN 63486=239 AND Y>0
```

```
120 DATA 0,0,1,66,68,40,16,12
130 DATA 96,8,20,34,66,128,0,0
140 DATA 0,0,128,66,34,20,8,96
150 DATA 0,129,129,129,193,243,235,255
160 DATA 255,129,129,129,129,129,255,255
170 DATA 255,129,129,129,129,129,129,255
180 DATA 255,24,24,24,24,24,255,255
190 DATA 24,60,126,126,126,126,126,126
200 DATA 0,124,66,66,124,64,64,0
210 DATA 0,60,66,66,82,74,60,0,0
220 DATA 124,66,66,124,68,66,0,0,60
```

```
): LET A$=("E" AND Y1>Y)+("F"
AND Y1<Y): IF Y1<>Y THEN
LET UN=UN+.25: GO TO 90
60 LET UN=UN+.15: LET A$="A"
70 IF ATTR (X+1,Y)=57 AND IN 61438=
239 THEN LET X=X+1: LET F=1
80 IF ATTR (X-1,Y)=57 AND IN 61438=
247 THEN LET X=X-1: LET F=1
90 PRINT AT X1,Y1;" "
100 IF F=1 THEN PRINT AT X1,Y1;
```

di Marcello Spero

copiate il programma principale e salvatelo con Goto 9999. Il salvataggio avverrà in due sezioni, una dopo l'altra; non dimenticatevi di premere un tasto, dopo che la prima sezione sarà terminata. Verificate ambedue le sezioni, con Verify "": Verify "" Code, seguito da Enter.


Interessante è il metodo usato per tener nota del record; questo viene inserito, con una Poke, nell'area della grafica definibile (Udg), in modo da rimanere inalterato anche nel caso il programma venga fatto ripartire con Run.

Controlli

- 5 per andare a sinistra.
- 6 per andare in giù.
- 7 per andare in su.
- 8 per andare a destra.
- 9 per scavare a sinistra.
- 0 per scavare a destra.

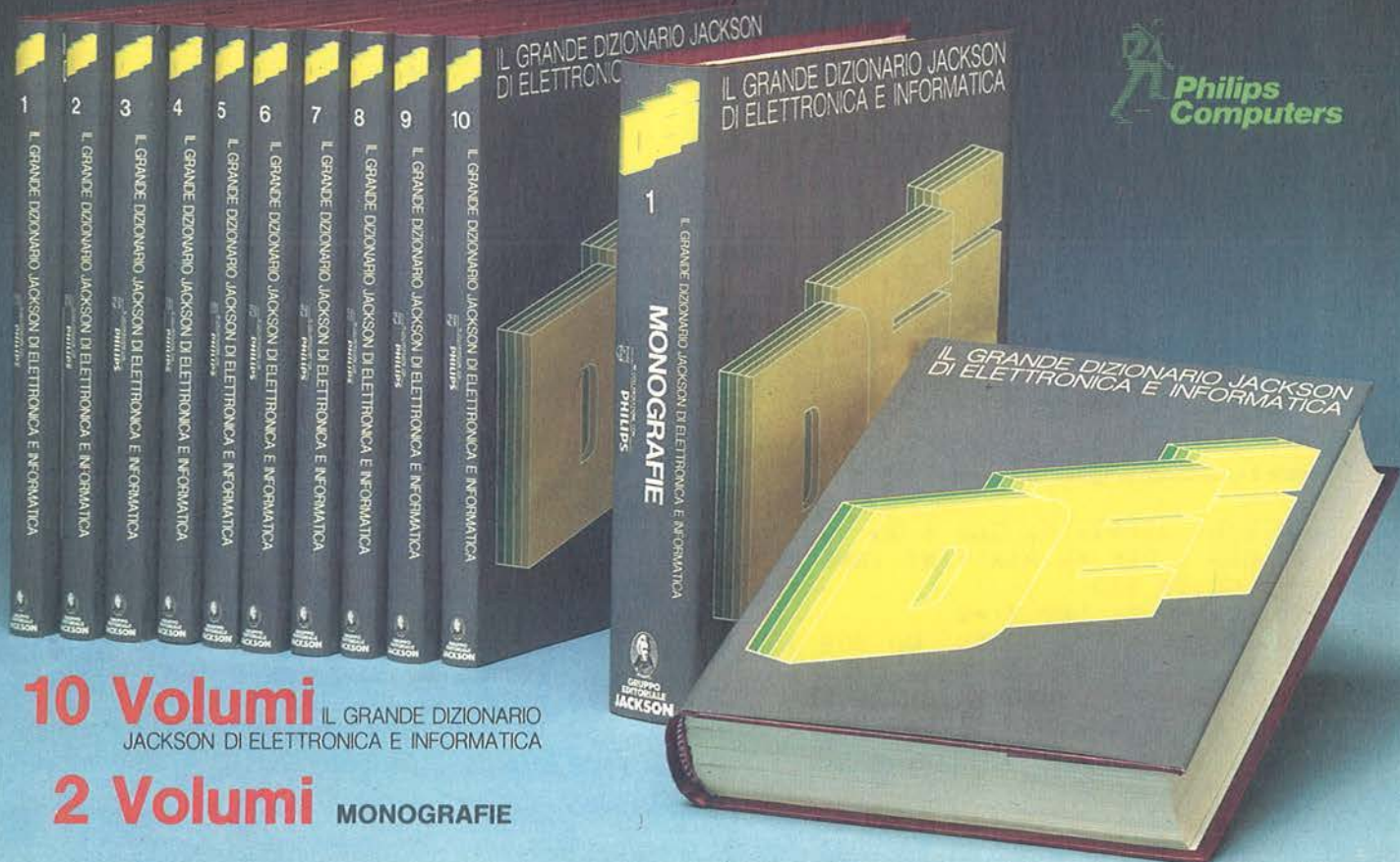
Le linee che contengono caratteri grafici sono le seguenti:

50, 60, 100, 150, 160, 170, 180, 200, 213, 300, 310, 380, 400, 405, 410, 470, 610, 620, 640, 720, 740, 780, 790, 800, 820. ■



IL GRANDE
DIZIONARIO
JACKSON
DI ELETTRONICA
E INFORMATICA

**LA SCIENZA DI OGGI
LA CULTURA DI DOMANI.**



Sinonimi inglesi e italiani
Traduzione italiana
Voce
Campo di appartenenza della voce
Definizione della voce
Rimando alla monografia
Le voci più importanti di elettronica, informatica e comunicazioni sono definite in modo esteso
Le illustrazioni (immagini o schemi) e le rispettive didascalie integrano, ampliano ed esemplificano la definizione della voce relativa
Acronimo

Rimando alla voce principale

La monografia tratta in modo esauriente e completo i concetti e le voci fondamentali

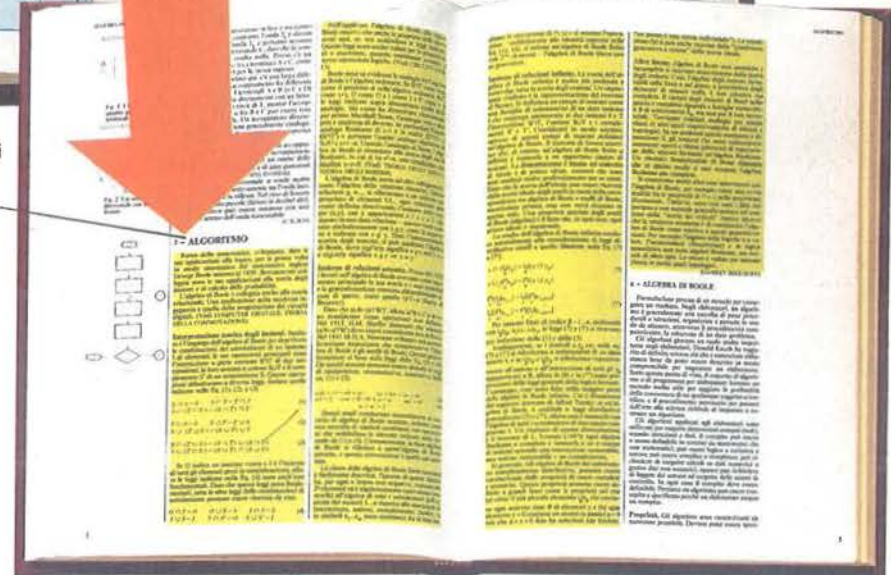


Il Grande Dizionario Jackson di Elettronica e Informatica

è il risultato di un importante lavoro di ricerca, durata più di cinque anni, che ha coinvolto decine di specialisti.

DEI è l'unica opera al mondo che risponde all'ormai improrogabile necessità di raccogliere e organizzare tutti i termini tecnici nati dalla diffusione massiccia dell'elettronica, informatica e comunicazione per formalizzare e consolidare il "nuovo sapere".

DEI è per tutti coloro che hanno capito l'importanza di sapersi orientare in questo nuovo universo linguistico; per chi vuole comprendere il significato di termini che le nuove discipline hanno introdotto anche nel linguaggio comune; per quanti desiderano saperli utilizzare con padronanza per non rimanere esclusi da un mondo che sta compiendo un cambiamento le cui dimensioni sono senza precedenti; per coloro, infine, che per lavoro, studio, informazione vogliono possedere, già da oggi, la "cultura di domani".



Con DEI, infatti, la trasformazione della scienza di oggi nella cultura di domani compie un decisivo passo avanti.

- DEI**
- 45000 termini
 - 3000 illustrazioni
 - 1000 monografie
- in 100 fascicoli settimanali da rilegare in 12 splendidi volumi.**

IN EDICOLA IL 1° E IL 2° FASCICOLO A SOLE L. 2.500



Seguito listato 2.

```

    INK 1;"D"
110 IF X1<>X THEN LET UN=UN+.15
120 LET F=0: IF ATTR (X,Y)=57 THEN
    LET F=1
130 IF IN 61438=255 OR X<>3 AND X
    <>8 AND X<>13 AND X<>18
    THEN GO TO 210
140 LET V=ATTR (X+1,Y+1): LET V1=
    ATTR (X+1,Y-1): IF V=58 OR V1=58
    THEN LET SC=SC+5: PRINT AT 0,6
    ;SC
150 IF IN 61438=253 AND ATTR (X-1,Y-
    1)<>57 AND ATTR (X-1,Y-1)<>6
    0 AND ATTR (X,Y-1)<>57 THEN
    LET UN=UN+.15: LET W=1: PRINT
    AT X,Y; INK 1;"E": FOR N=1 TO 5:
    PRINT AT X,Y-1;"G": PAUSE 2:
    PRINT AT X,Y-1;"H": PAUSE 2:
    PRINT AT X,Y-1;"I": BEEP .01,Y:
    PRINT AT X+1,Y-1;"K": NEXT N:
    PRINT AT X+1,Y-1;" "
160 IF IN 61438=254 AND ATTR (X,Y+1)
    <>57 AND ATTR (X-1,Y+1)<>57
    AND ATTR (X-1,Y+1)<>60 THEN
    LET UN=UN+.15: LET W=2: PRINT
    AT X,Y; INK 1;"F": FOR N=1 TO 5:
    PRINT AT X,Y+1;"I": PAUSE 2:
    PRINT AT X,Y;"J": PAUSE 2:
    PRINT AT X,Y+1;" ": BEEP .01,Y:
    PRINT AT X+1,Y+1;"K": NEXT N:
    PRINT AT X+1,Y+1;" "
170 IF V1=58 AND W=1 THEN PRINT
    AT X+1,Y-1; PAPER 6;"C"
180 IF V=58 AND W=2 THEN PRINT AT X
    +1,Y+1; PAPER 6;"C"
190 LET W=0
200 IF ATTR (X,Y)=60 THEN PRINT
    AT 2,5;"SEI STATO DIVORATO!";
    AT X,Y; INK 1; OVER 1; FLASH 1;"
    A": FOR N=0 TO 100: NEXT N:
    GO TO 530
210 IF ATTR (X,Y)=62 THEN GO TO 880
213 PRINT AT X,Y; INK 1;A$: IF
    ATTR (X+1,Y)=56 THEN PRINT AT 2
    ,2;"SEI CADUTO IN UNA BUCA";AT X
    ,Y;" "; INK 1;AT X+1,Y;"A":
    FOR N=1 TO 100: NEXT N: GO TO 5
    30
215 IF XY<=125 AND F2=0 THEN
    GO TO 850
220 FOR N=1 TO (UN*10)/LV: PLOT
    INK 2;XY,8: DRAW INK 2; OVER 1
    ;0,7: LET XY=XY-1: LET UN=UN-.1:
    NEXT N
230 IF XY<=72 THEN PRINT AT 2,0;"SE

```

```

    I MORTO ASFISSIATO": FOR N=1
    TO 100: NEXT N: GO TO 530
240 LET A1=A(Z): LET B1=B(Z)
250 IF ATTR (A1+1,B1)=56 THEN
    GO TO 470
260 IF A1=X THEN IF A1=3 OR A1=8
    OR A1=13 OR A1=18 THEN GO TO 3
    50
270 IF C(Z)<>0 THEN GO TO 300
280 IF ATTR (A1+1,B1)=57 OR ATTR (A1
    -1,B1)=57 THEN GO TO 330
290 GO TO 350
300 LET A(Z)=A(Z)+C(Z): IF A(Z)=3
    OR A(Z)=8 OR A(Z)=13 OR A(Z)=18
    THEN LET C(Z)=0: PRINT AT A1,B
    1; INK 1;"D": LET A1=A(Z): LET F
    (Z)=1: GO TO 350
310 PRINT AT A1,B1; INK 1;"D": LET A
    1=A(Z): LET B1=B(Z): GO TO 400
320 GO TO 400
330 IF A1<X AND ATTR (A1+1,B1)=57
    THEN LET C(Z)=1: GO TO 300
340 IF A1>X AND ATTR (A1-1,B1)=57
    THEN LET C(Z)=-1: GO TO 300
350 IF A1=X THEN LET B1=B1+(B1<Y)-
    (B1>Y): GO TO 380
360 LET B1=B1+D(Z)
370 IF B1>=31 OR B1<=0 THEN LET D(Z)
    =-D(Z)
380 PRINT AT A(Z),B(Z);" ": IF F(Z)=
    1 THEN PRINT AT A(Z),B(Z);
    INK 1;"D"
390 LET F(Z)=0: IF ATTR (A1,B1)=57
    THEN LET F(Z)=1
400 PRINT AT A1,B1; INK 4;"B"
405 IF F2=1 THEN PRINT AT F5,F4;
    OVER 1; INK 6;"O"
410 IF A1=X AND Y=B1 THEN PRINT
    AT 2,5;"SEI STATO DIVORATO";AT X
    ,Y; OVER 1; INK 1; FLASH 1;"A":
    FOR N=0 TO 100: NEXT N: GO TO 5
    30
420 IF ATTR (A1+1,B1)=56 THEN
    GO TO 470
430 LET A(Z)=A1: LET B(Z)=B1
440 LET Z=Z+1: IF Z>=C1 THEN LET Z=
    1
450 IF INKEY$="H" THEN GO SUB 750
460 GO TO 30
470 PRINT AT A1,B1;" ";AT A1+1,B1;
    INK 2;"B": FOR N=1 TO 4: BEEP
    1,N: BEEP .1,5-N: NEXT N
480 IF C1-1=1 THEN LET SC=SC+10:
    GO TO 520
490 LET A(Z)=A(C1-1): LET B(Z)=B(C1-
    1): LET F(Z)=F(C1-1)

```


ZX SPECTRUM

Seguito Ilistato 2.

```

500 LET SC=SC+10: LET C1=C1-1
510 PRINT AT 0,6;SC: LET Z=1:
    GO TO 440
520 PRINT AT 1,0; FLASH 1; INK 4;"BO
    NUS": FOR N=XY TO 72 STEP -1:
    BEEP .01,XY/4: PLOT XY,8:
    DRAW INK 2; OVER 1;0,7: LET SC
    =SC+1: PRINT AT 0,6;SC: LET XY=X
    Y-1: NEXT N: FOR N=1 TO 50:
    BEEP .05,N: BEEP .05,50-N:
    NEXT N: GO TO 20
530 IF SC>HI THEN PRINT AT 1,0;
    FLASH 1; INK 2;"          NUOVO RE
    CORD!!!          ": FOR N=0 TO 5
    0: BEEP .01,N: BEEP .01,25-N:
    NEXT N: POKE USR "U",SC-256*
    INT (SC/256): POKE USR "U"+1,
    INT (SC/256): GO TO 550
540 PRINT AT 0,0;: FOR N=1 TO 11:
    PRINT INK 7; PAPER 8; OVER 1;
    FLASH 1;"
          ": NEXT N: FOR N=1
    TO 2: BEEP .5,-5: BEEP .5,-10:
    PAUSE 40: NEXT N: FOR N=50 TO 0
    STEP -1: BEEP .05,N: NEXT N
550 PRINT AT 11,3;"PREMI UN TASTO PE
    R INIZIARE";AT 13,3;"PREMI 'I' P
    ER LE ISTRUZIONI": IF IN 254=255
    THEN GO TO 550
570 GO TO (575 AND (INKEY$<>"i"
    AND INKEY$<>"I"))+(780 AND (
    INKEY$="i" OR INKEY$="I"))
575 INPUT "LIVELLO (1-5)"; LINE L$:
    IF L$>"5" OR L$<"1" THEN
    GO TO 575
576 LET LV=(VAL L$-0.9)*5: GO TO 600
580 LET N=0: POKE USR "U",N-256*
    INT (N/256): POKE USR "U"+1,
    INT (N/256)
590 BORDER 5: PAPER 7: CLS : PRINT
    INK 0;AT 13,3;"PREMI 'I' PER LE
    ISTRUZIONI": FOR N=19 TO 21:
    PRINT AT N,0; INK 4;"{32SG8}":
    NEXT N: PRINT AT 20,1; INK 2;"P
    A N I C O S P E T T R A L E":
    PRINT AT 11,3;"PREMI UN TASTO P
    ER INIZIARE": PAUSE 0: GO TO 570
600 CLS : LET LE=0: LET SC=0: LET HI
    =PEEK USR "U"+256*PEEK (USR "U"+
    1)
610 INK 7: CLS : INK 0: PRINT AT 19,
    0; PAPER 6;"CCCCCCCCCCCCCCCCCCCC

```

```

CCCCCCCCCCCC"; INK 2; PAPER 7;"O
SSIGENO>"; INK 0; PAPER 6;"MMMMM
MMMMMMMMMMMMMMMMMMMMNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNN"
620 FOR N=4 TO 14 STEP 5: PRINT AT N
,0; PAPER 6;"CCCCCCCCCCCCCCCCCCCC
CCCCCCCCCCCCCCCCLLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLLL": BEEP .1,50-N:
    BEEP .1,N: NEXT N
630 PAUSE 25
640 FOR N=3 TO 13 STEP 5: LET Z=FN R
(13)+1: FOR F=0 TO 5: PRINT AT N
+F,Z; INK 1;"D": BEEP .05,N+F:
    PRINT AT N+F,31-Z; INK 1;"D":
    BEEP .05,N+F: NEXT F: BEEP .1,F
    : PAUSE 10: NEXT N
650 PRINT AT 0,0;"PUNTI>";SC;AT 0,16
;"RECORD>";HI
660 LET Y=FN R(13)+1: LET W=FN R(4):
    IF W=0 THEN LET X=3
670 IF W=1 THEN LET X=8
680 IF W=2 THEN LET X=13
690 IF W=3 THEN LET X=18
700 IF ATTR (X,Y)=57 THEN GO TO 660

710 LET F2=0: LET Z=1: DIM D(4):
    DIM A(4): DIM B(4): DIM C(4)
720 LET A(1)=2: LET D(1)=1: LET B(1)
    =FN R(30)+1: PRINT AT A(1),B(1);
    INK 4;"B": BEEP .01,A(1): FOR N
    =2 TO 4: LET A(N)=A(N-1)+5:
    LET D(N)=1
730 LET B(N)=FN R(30)+1: IF ATTR (A(
    N),B(N))=57 OR A(N)=X AND B(N)=Y
    THEN GO TO 730
740 LET W=0: LET C1=5: PRINT AT A(N)
,B(N); INK 4;"B": BEEP .01,A(N):
    NEXT N: LET LE=LE+1: DIM F(4):
    LET UN=0: LET XY=(30-LE)*8+7:
    FOR N=9 TO 30-LE: BEEP .01,N:
    PRINT AT 20,N; INK 2;"{SG8}":
    NEXT N: GO TO 30
750 PRINT AT 1,12; FLASH 1; INK 2;"G
    IOCO "; INK 1;"BLOCCATO": IF
    IN 25=255 OR INKEY$="H" THEN
    GO TO 750
760 PAUSE 0: PRINT AT 1,12;"
    ": RETURN
770 POKE 23658,8: CLS : PRINT AT 11,
    5; INK 1; FLASH 1;"CARICAMENTO";
    INK 2;"CARICAMENTO";AT 12,5;"CA
    RICAMENTO"; INK 1;"CARICAMENTO":
    INK 7: LOAD "udg"CODE USR "A":
    INK 0: GO TO 580
780 CLS : PRINT "*ISTRUZIONI*" "USA

```



Dalla grande edicola Jackson

Tutto sull'hobby e home computer

VIDEO Giochi & COMPUTER

La guida indiscussa al fantastico mondo dei videogames. La più eccitante, divertente, istruttiva rassegna del settore.

11 numeri all'anno: L. 4.000 a numero
Abbonamento: solo L. 33.000

PERSONAL SOFTWARE

Aspetti e problemi del software per personal computer, programmi, giochi e sistemi operativi.

11 numeri all'anno: L. 4.500 a numero
Abbonamento: solo L. 39.000

ELETRONICA

Il nuovo fai da te per l'hobbista elettronico, con progetti sempre nuovi e originali da realizzare e divagazioni sul mondo dei micro computer.

12 numeri all'anno: L. 3.500 a numero
Abbonamento: L. 32.000

strumenti MUSICALI

Il mondo delle 7 note in versione ... elettronica. Con test strumentali, novità e analisi del mercato, servizi speciali.

11 numeri all'anno: L. 4.000 a numero
Abbonamento: solo L. 35.000

Quando l'informazione fa testo

In busta chiusa inviate questo coupon a:

Gruppo Editoriale Jackson via Rosellini, 12 - 20124 MI

Desidero ricevere GRATIS un numero della Rivista

(allego L. 1.000 in francobolli per contributo spese di spedizione)

Inviatemi GRATIS il Catalogo della Biblioteca JACKSON (allego L. 1.000 in francobolli per contributo spese di spedizione)

Nome _____

Cognome _____

via _____

CAP _____ Città _____

ZX SPECTRUM

Seguito listato 2.

```

I TASTI CURSORE PER MUOVERTI""9
E 0 PER SCAVARE,"""H PER BLOCCA
RE IL GIOCO,""" FLASH 1""SCAVO":
FOR N=0 TO 31: PRINT PAPER 6;
AT 12,N;"C";AT 13,N;"L": NEXT N
790 PRINT AT 11,16; INK 1;"E": FOR N
=1 TO 5: PRINT AT 11,15;"G":
PAUSE 2: PRINT AT 11,15;"H":
PAUSE 2: PRINT AT 11,15;"-":
BEEP .01,11: PRINT AT 12,15;"K"
: NEXT N: PRINT AT 12,15;" "
800 PAUSE 50: PRINT AT 7,0; FLASH 1;
"ALIENO: 10 PUNTI": FOR N=1 TO 1
4: PRINT AT 11,N; INK 4;" B":
PAUSE 15: NEXT N: PRINT AT 11,1
5;" ";AT 12,15; INK 2;"B": FOR N
=1 TO 4: BEEP .1,N: BEEP .1,5-N:
NEXT N
810 PAUSE 50: PRINT AT 7,0; FLASH 1;
"ALIENO SEPOLTO: 5 PUNTI"
820 PRINT AT 11,16; INK 1;"E": FOR N
=1 TO 5: PRINT AT 11,15;"G":
PAUSE 2: PRINT AT 11,15;"H":
PAUSE 2: PRINT AT 11,15;"-":
BEEP .01,11: PRINT AT 12,15;"K"
: NEXT N: PRINT AT 12,15;
PAPER 6;"C"
830 PRINT AT 15,0;"PREMI 'I' PER RIP
ETERE""LE ISTRUZIONI""O UN AL
TRO TASTO PER INIZIARE": PAUSE 7
5
840 GO TO (780 AND INKEY$="I")+10
AND INKEY$<>"I")+830 AND
INKEY$=""
850 IF RND>.7 THEN LET F2=2:
GO TO 220
860 PRINT AT 1,10; INK 6; FLASH 1;"O
SSIGENO DI RISERVA": LET F4=
INT (RND*32): LET F6=FN R(4):
LET F5=(3 AND F6=0)+(8 AND F6=1
)+(13 AND F6=2)+(18 AND F6=3)
865 IF ATTR (F5,F4)=57 THEN GO TO 8
60
870 LET f2=1: GO TO 220
880 PRINT AT 1,10;"
": LET f2=2: LET f3=INT (RND*4
0)+30: FOR n=1 TO f3: LET xy=xy+
1: PLOT xy,8: DRAW INK 2;0,7:
BEEP .01,xy/5: NEXT n
890 GO TO 213
9998 STOP
9999 SAVE "panico" LINE 770: SAVE "ud
g"CODE USR "a",146

```

La gestione del proprio patrimonio di riviste, che consenta un'immediato reperimento delle informazioni ordinate per argomento, è probabilmente quanto di meglio possiamo fare con il nostro personal.

Quante volte infatti ci siamo persi nel cercare qualcosa di già visto pubblicato su periodici, arenandoci regolarmente nello sfogliare un mare di riviste senza trovare ciò che ci interessa? Ovviamente per poter ottenere qualche risultato bisogna che siano soddisfatte alcune condizioni.

Anzitutto la lettura di ciascun fascicolo dovrà tassativamente terminare con l'archiviazione dei contenuti su un supporto magnetico; inoltre è necessario disporre in un tempo sufficientemente breve del materiale archiviato e questo può avvenire solo disponendo di un buon programma di archiviazione che si adatti nel migliore dei modi all'applicazione prevista.

Il software di base fornito (gratuitamente) con il QL contiene il programma Archive, un data base molto potente con un ampio set di istruzioni specializzate nella manipolazione di file di dati e la possibilità di creare procedure ad hoc con un linguaggio decisamente simile più al Pascal che al Basic.

Il maggior rischio che si corre archiviando una raccolta di articoli è rappresentato dalla disomogeneità delle informazioni in essi contenuti; non ha infatti nessun senso raccogliere semplicemente dei riassunti se poi dobbiamo scorrere tutto l'archivio per trovare ciò che cerchiamo.

La prima idea è quella di raccogliere in campi specifici alcuni aspetti salienti del pezzo in questione (i cosiddetti campi guida); purtroppo il più delle volte questo non basta, perché esiste sempre il rischio di memorizzare informazioni riguardo ad argomenti analoghi sotto voci sinonime, che sfuggono alla successiva ricerca mirata.

È quindi preferibile che il computer stesso guidi la classificazione, proponendo un insieme di parole-chiave cui fare costante riferimento.

Se l'applicazione è poco meno che banale, è una buona idea strutturare questo dizionario in modo che termini di orizzonte più vasto consentano di accedere a sottoliste di argomenti più speci-

Emeroteca con "Archive"

Archiviamo con intelligenza
gli articoli delle riviste

di Giovanni Tisi

Listato 1 - Listato procedura "Emer-Prg".

```

proc aggiungi
  use "a"
  let altri=1
  while altri
    sinput sogg$,macco$,rif$,note$
    scan
    use "a"
    let code$=per$
    append
    sprint
    print at 13,0; "ALTRI ?(Y/N)"
    let altri=upper(getkey())="Y"
  endwhile
endproc

proc apri
  open "rivi"logical "a"
  look "thes"logical "t"
  sload "ts"
  while memory()
    print at 13,0; " Aggiungi Ricerca Correggi"
    let a$=upper(getkey())
    if a$="A": aggiungi: endif
    if a$="R": ricerca: endif
    if a$="C": correggi: endif
  endwhile
endproc

proc ricerca
  print at 13,0; "Premere T per avvlersi del catalogo per argomenti"
  while upper(getkey())="T"
    scan: use "a"
    reset
    let l=len(per$)
    select code$( to l)=per$
    print at 13,0; "Esistono "; count(); " records con questo codice. T per pro
  seguire"
  endwhile
endproc

proc scan
  let per$=""
  let a$=""
  let anc=1
  use "t": first
  while nip and anc
    search no$=per$
    print at 13,0; "
    "
    print at 13,0; fig$
    let a$=upper(getkey())
    let anc=code(a$)>10
    if anc: let per$=per$+a$: endif
  endwhile
  print at 13,0; "
endproc

```

SINCLAIR QL

fici, fino a giungere al massimo grado di definizione richiesta dal problema.

Questo dizionario dovrebbe essere facilmente modificabile, a esempio per consentire l'introduzione di nuovi termini prima sottovalutati o per l'eliminazione di doppietti, e la sua scansione deve dare come risultato un codice alfabetico o numerico che indichi in modo univoco il percorso seguito dalla radice di ingresso al nodo finale.

Si noti che in fase di ricerca è spesso assai utile non essere troppo specifici nella richiesta, a esempio può servire un elenco di tutte le periferiche grafiche, non solo delle stampanti o dei plotter, quindi è bene prevedere che la scansione del dizionario possa essere interrotta in qualsiasi momento.

In pratica l'insieme delle parole chiave viene memorizzato in una struttura ad albero un po' particolare, dove ogni nodo è rappresentato da un record contenente tutte le possibili scelte per proseguire verso il basso.

Ogni record viene raggiunto durante la

scansione perché provvisto di un campo iniziale che funge da target, contenendo l'esatto cammino che ha portato dalla radice fino a lui.

Per risparmiare memoria i record terminali, cioè quelli che non avrebbero successori, non vengono creati; il fatto è segnalato in un apposito campo.

Questa lunga chiacchierata può aver scoraggiato qualcuno, ma possono ricredersi osservando che un thesaurus (così viene chiamato in modo aulico il dizionario) già abbastanza completo, che consenta la catalogazione di riviste di informatica, può essere agevolmente rappresentato da una quindicina di record e alloggiare quindi permanentemente in memoria.

Riempimento del dizionario

La dozzina di record di Thes va caricata manualmente secondo uno schema predefinito che risponda alle particolari esigenze di ognuno; in figura 1 e 2 è

mostrato il thesaurus 7 utilizzato per riviste di informatica, ma è meglio che ciascuno studi un suo albero di catalogazione.

Procedure

Per utilizzare in modo proficuo l'archivio sono poi necessarie alcune routine come quelle mostrate nel listato.

Anzitutto va disegnato l'algoritmo di scansione del thesaurus, che, partendo dall'inizio, consenta di scendere fino alla profondità desiderata, restituendo nella variabile globale Per\$ il percorso seguito; verrà utilizzata sia per catalogare che per selezionare in fase di ricerca. Oltre a questa sono mostrate, a puro titolo d'esempio, procedure per l'aggiunta di articoli e per la ricerca degli stessi via Code\$. Tutto l'insieme è coordinato dalla routine Apri e fa riferimento a un layout dello schermo memorizzato in un file TS-Scn che ognuno potrà personalizzare a proprio piacimento. ■

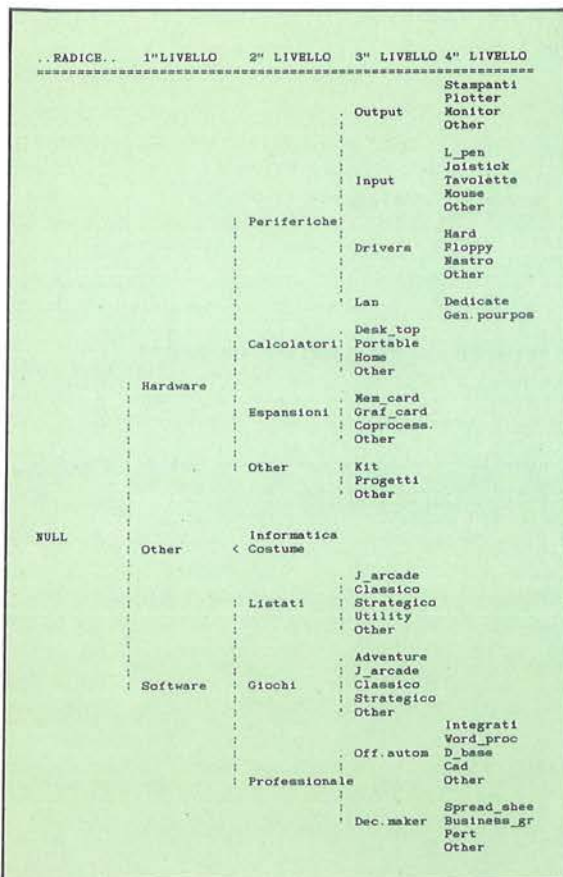


Figura 1 - L'albero di classificazione.

```

no$      : HP
fig$     : Output Input Drivers Lan
nip      : 1
=====
no$      : HPO
fig$     : Stampanti Plotters Monitors Other
nip      : 0

sogg$    : Mac Writer
macc$    : Mac Intosh
rif$     : Bit 54
note$    : prova completa
code$    : SPAI

```

Figura 2 - Record di "Thes" e di "Rivi".

Creazione dell'archivio

Create "Rivi" Archivio dei riassunti.
 Sogg\$ Soggetto dell'articolo.
 Macc\$ Macchina.
 Rif\$ Riferimento alla fonte.
 Note\$ Contenuto.
 Code\$ Contiene il percorso di identificazione ed è quindi univoco per argomento.

Endcreate

Create "Thes" Dizionario strutturato delle parole-chiave.
 No\$ Target di identificazione.
 Fig\$ Possibili opzioni per proseguire.
 Nip Flag di fine scansione.

Endcreate

Questo programma si rivolge alle persone appassionate di cruciverba, che vogliono provarlo su computer e a quelle alle quali il cruciverba non piace molto (o non piace affatto), che però apprezzando i videogame, desiderino cimentarsi almeno una volta.

Il gioco

Dopo avere dato il Run, apparirà la solita presentazione. Dopo avere premuto un tasto, avremo il menu. Abbiamo due opzioni.

1. Regole del gioco

Con questa prima opzione, potremo vedere le istruzioni che riguardano il gioco e l'utilizzazione del programma. Adesso, ne elencheremo qualcuna:

1) quando digiteremo una definizione, accorgendoci che questa è sbagliata, non potremo correggerla immediatamente: dovremo richiamarla in seguito; 2) quando il programma ritiene che abbiamo completato il cruciverba, ci chiede se vogliamo smettere oppure no. In caso affermativo, sapremo quante definizioni orizzontali e quante verticali avremo indovinato. Se non vogliamo terminare, allora il computer ci darà ancora dieci tentativi per completare il cruciverba, continuando a darceli fino a quando non gli diremo che vogliamo smettere.

2. Il gioco

Selezionando questa opzione, entreremo nel vivo del gioco. Immediatamente ci verrà chiesto quale cruciverba desideriamo (ce ne sono registrati tre). Dopo avere atteso un attimo ci verrà chiesto se vogliamo le definizioni orizzontali o verticali. Scelta una delle due, ci verrà fornito il numero delle scelte. Dopo averne selezionata una, avremo la definizione e, poco dopo, potremo scrivere quella che ci sembra la risposta esatta.

Dopo aver battuto il tasto Return, vedremo il reticolato classico dei cruciverba e la risposta sarà stampata al suo interno. Dopo poco comparirà il messaggio "Batti un tasto". Dopo averlo premuto il programma cancellerà il reticolato e tornerà a chiederci se vogliamo una definizione orizzontale o verticale, altrimenti potremo studiare il cruciverba senza alcuna limitazione di tempo.

Cruci Msx

Ovvero il primo esempio di cruciverba su computer Msx

di Luciano Cassi

Variabili e vettori principali

Variabili

AA, BB, CC, DD, XX, - Contatori. AA e BB sono rispettivamente il numero delle definizioni orizzontali e verticali. XX è il contatore che, sommando i valori di AA e BB, confrontato poi con un altro contatore, dice al programma quando il giocatore dovrebbe avere completato il cruciverba. CC e DD sono utilizzati al termine. Questi vengono incrementati solamente se le definizioni orizzontali o verticali sono esatte.

Vettori

DO\$, DV\$, D1\$, D2\$ - Questi vettori contengono rispettivamente: le definizioni orizzontali, quelle verticali, le risposte delle definizioni orizzontali e delle verticali.

A\$, A1\$, -Vengono utilizzati, il primo per le orizzontali, l'altro per le verticali, entrambi nelle due routine per l'incolonnamento delle risposte nel reticolato.

RV, RO, K, R1, R2, K1, - Questi vettori numerici servono per: posizionamento orizzontale, verticale e lunghezza della risposta orizzontale, posizionamento orizzontale, verticale e lunghezza della definizione verticale.

Commento al listato

10 - Inizializzazione.
20 - Azzeramento variabili e dimensionamento vettori.
30 - Pulizia schermo.
40 - Stampa menu.
50 - Input B\$; trasformazione del dato stringa in dato numerico, controllo se il

dato numerico è compreso tra uno e due.

60 - A seconda del valore del dato numerico, l'esecuzione del programma passa o al sottoprogramma sito dalla riga 1000, o a quello sito dalla riga 5000.

70 - Se il valore di W è uno, torna alla riga 30.

80-100 - Cambio colori e modo dello schermo, ripristino delle key e fine programma.

1000-1140 - Spiegazioni e uso del programma.

5000 - Pulizia schermo.

5010 - Stampa il menu.

5020 - Esecuzione uguale alla riga 50.

5030 - A seconda del valore di B, il programma va a caricare le date relative al cruciverba scelto.

5040 - Pulizia schermo.

5050-5070 - Routine per scegliere da quale gruppo si vuole la definizione. Se si vogliono le definizioni orizzontali andremo al sottoprogramma sito dalla riga 7000.

5110-5480 - Disegno del reticolato grafico.

5490-5540 - Routine per il posizionamento sul reticolato delle risposte orizzontali.

5550-5590 - Routine per il ripristino dei vettori che contengono i dati per il posizionamento delle risposte orizzontali.

5600-5650 - Esecuzione uguale alla routine sita da 5490-5540. Adattamento per le risposte verticali.

5660-5700 - Esecuzione identica alla routine da 5550-5590. Attuate le stesse modifiche della routine da 5600-5650.

5710 - Stampa la frase "Batti un tasto".

5720-5730 - Input B\$. Se il valore di TC è uguale a XX, incrementane il valore di dieci e vai a 5740.

5740-5770 - Ci viene chiesto se voglia-

mo continuare a giocare. Se si, torniamo l'esecuzione alla riga 5040.

5780 - Azzeramento contatori CC,DD.

5790-5820 - Routine per definire quante definizioni orizzontali sono state indovinate.

5830-5860 - Routine per definire quante definizioni verticali sono state indovinate.

5870-5910 - Stampa dei risultati e ritorno alla riga 70.

6000-6180 - Subroutine per la scelta, la

visione e l'input della definizione orizzontale.

7000-7180 - Subroutine per la scelta, la visione e l'input della definizione verticale.

15000-19400 - Data per i cruciverba. ■

Listato 1 - Il programma Cruci Msx.

```
10 CLS:KEYOFF: CLEAR 3000:MAXFILES=1:GOSU
B 10000
20 KEYOFF: DIM D0$(40): DIM DV$(40): DIM D1
$(40): DIM D2$(40): DIM A$(40): DIM A1$(40)
: DIM RV(40): DIM RO(40): DIM R1(40): DIM R2
(40): DIM K(40): DIM K1(40)
30 LINE(0,0)-(255,192),1,BF: COLOR,1,1: LI
NE(0,0)-(255,192),11,B
40 PSET(60,80),1: COLOR4: PRINT#1,"1)REGOL
E DEL GIOCO": PSET(60,100),1: COLOR6: PRINT
#1,"2)IL GIOCO": B$=""
50 B$=INKEY$: W=VAL(B$): IF W<1 XOR W>2 TH
EN 50
60 ON W GOSUB 1000,5000
70 IF W=1 THEN 30
80 COLOR15,4,4
90 SCREEN0:KEYON
100 END
1000 LINE(0,0)-(255,192),1,BF
1010 PSET(48,10),1: COLOR11: PRINT#1,"Benv
enuti a ";: COLOR9: PRINT#1,"CRUCI/MSX"
1020 PSET(8,30),1: COLOR11: PRINT#1,"Scopo
di CRUCI/MSX è di farvi divertire con
uno dei giochi enigmistici più belli
";: COLOR5: PRINT#1,"IL CRUCI-
VERBA."
1030 PSET(8,70),1: COLOR11: PRINT#1,"All'i
nterno del programma ve ne sono registra
ti 3.": PSET(8,90),1: PRINT#1,"Dopo averne
selezionato uno,vi verrà chiesto se vo
lete una de- finizione orizzontale o ver
ti- cale."
1040 PSET(8,130),1: PRINT#1,"Dopo avere d
igitato il numero della definizione oh
e volete, questa verrà visualizzata e
po- trete scrivere la vostra rispo- sta.
"
1050 PSET(72,184),1: COLOR5: PRINT#1,"BATT
I UN TASTO": B$=""
1060 B$=INKEY$: IF B$="" THEN 1060
1070 LINE(0,0)-(255,192),1,BF
1080 PSET(88,10),1: COLOR6: PRINT#1,"ATTEN
ZIONE": PSET(8,30),1: COLOR11: PRINT#1,"1)Q
uando digitate il numero per scegliere l
a definizione,se è inferiore a 10,dovr
à essere di- gitato così:01,02,03 ec."
1090 PSET(8,70),1: COLOR5: PRINT#1,"2)Se s
bagliate a scrivere una definizione,no
n potrete correg- gerla subito,dovrete r
```

```
chiede- rla in un secondo momento."
1100 PSET(8,110),1: COLOR2: PRINT#1,"3)Sia
quando chiedete una defi- nizione,sia q
uando rispondete a quest'ultima,fatelo i
n MAIUSCO- LO."
1110 PSET(8,150),1: COLOR14: PRINT#1,"4)Ad
un certo punto vi sarà chiesto se v
orrete continuare a giocare.Se non vorre
te,saprete le definizioni indovinate."
1120 PSET(72,184),1: COLOR5: PRINT#1,"BATT
I UN TASTO": B$=""
1130 B$=INKEY$: IF B$="" THEN 1130
1140 RETURN
5000 LINE(0,0)-(255,192),1,BF
5010 PSET(70,80),1: COLOR11: PRINT#1,"1=CR
UCIVERBA 1": PSET(70,100),1: COLOR9: PRINT#
1,"2=CRUCIVERBA 2": PSET(70,120),1: COLOR4
: PRINT#1,"3=CRUCIVERBA 3": B$=""
5020 PSET(70,160),1: B$=INKEY$: COLOR2: PRI
NT#1,"SCEGLI ";: B$: JK=VAL(B$): IF JK<1 XOR
JK>3 THEN LINE(120,160)-(255,170),1,BF:
GOTO 5020
5030 ON JK GOSUB 15000,17000,19000
5040 LINE(0,0)-(255,192),1,BF
5050 PSET(52,60),1: COLOR5: PRINT#1,"Vuoi
le definizioni": COLOR6: PSET(84,87),1: PRI
NT#1,"ORIZZONTALI": COLOR11: PSET(124,104)
,1: PRINT#1,"0": COLOR2: PSET(91,121),1: PRI
NT#1,"VERTICALI"
5060 COLOR14: PSET(108,134),1: PRINT#1,"(O
>V)": B$=""
5070 B$=INKEY$: IF B$="" THEN 5070 ELSE IF
B$="0" THEN GOSUB 6000 ELSE IF B$<>"V" TH
EN 5070 ELSE GOSUB 7000
5100 LINE(0,0)-(255,192),1,BF
5110 COLOR 6
5120 LINE (24,24)-(39,154),,B
5130 LINE (40,24)-(55,154),,B
5140 LINE (56,24)-(71,154),,B
5150 LINE (72,24)-(87,154),,B
5160 LINE (88,24)-(103,154),,B
5170 LINE (104,24)-(119,154),,B
5180 LINE (120,24)-(135,154),,B
5190 LINE (136,24)-(151,154),,B
5200 LINE (152,24)-(167,154),,B
5210 LINE (168,24)-(183,154),,B
5220 LINE (184,24)-(199,154),,B
5230 LINE (200,24)-(215,154),,B
5240 LINE (216,24)-(231,154),,B
5250 PSET (24,34)
5260 LINE -(231,34)
```

Seguito listato Cruci Msx.

```

5270 PSET (231,44)
5280 LINE -(24,44)
5290 PSET (24,54)
5300 LINE -(231,54)
5310 PSET (231,64)
5320 LINE -(24,64)
5330 PSET (24,74)
5340 LINE -(231,74)
5350 PSET (231,84)
5360 LINE -(24,84)
5370 PSET (24,94)
5380 LINE -(231,94)
5390 PSET (231,104)
5400 LINE -(24,104)
5410 PSET (24,114)
5420 LINE -(231,114)
5430 PSET (231,124)
5440 LINE -(24,124)
5450 PSET (24,134)
5460 LINE -(231,134)
5470 PSET (231,144)
5480 LINE -(24,144)
5485 IF JK=2 THEN LINE(24,84)-(39,94),6,
BF:LINE(24,104)-(39,114),6,BF
5490 FOR I=1 TO 40
5500 IF A$(I)="" THEN 5540
5510 D=D+1:A2$=MID$(A$(I),D,1):PSET(RV(I),
R0(I)),1:COLOR4:PRINT#1,A2$
5520 IF D=K(I) THEN 5530 ELSE IF D<K(I) TH
EN RV(I)=RV(I)+16:GOTO 5510
5530 IF RV(I)>=222 THEN 5540 ELSE RV(I)=
RV(I)+9:R0(I)=R0(I)-2:LINE(RV(I),R0(I))-
(RV(I)+15,R0(I)+10),6,BF
5540 D=0:NEXT I
5550 FOR I=1 TO 40
5560 IF A$(I)="" THEN 5590
5570 IF RV(I)>=222 THEN 5580 ELSE RV(I)=
RV(I)-9:R0(I)=R0(I)+2
5580 T=16*(K(I)-1):RV(I)=RV(I)-T
5590 T=0:NEXT I
5600 FOR I=1 TO 40
5610 IF A1$(I)="" THEN 5650
5620 D=D+1:A3$=MID$(A1$(I),D,1):PSET(R1(
I),R2(I)),1:COLOR4:PRINT#1,A3$
5630 IF D=K1(I) THEN 5640 ELSE IF D<K1(I)
THEN R2(I)=R2(I)+10:GOTO 5620
5640 IF R2(I)=146 THEN 5650 ELSE R1(I)=
R1(I)-6:R2(I)=R2(I)+8:LINE(R1(I),R2(I))-
(R1(I)+15,R2(I)+10),6,BF
5650 D=0:NEXT I
5660 FOR I=1 TO 40
5670 IF A1$(I)="" THEN 5700
5680 IF R2(I)=146 THEN 5690 ELSE R1(I)=
R1(I)+6:R2(I)=R2(I)-8
5690 T=10*(K1(I)-1):R2(I)=R2(I)-T
5700 T=0:NEXT I
5710 PSET(72,180),1:COLOR11:PRINT#1,"BAT

```

```

TI UN TASTO"
5720 B$=INKEY$:IF B$="" THEN 5720
5730 TC=TC+1:IF TC<XX THEN 5040 ELSE IF
TC=XX THEN XX=XX+10:GOTO 5740
5740 LINE(0,0)-(255,192),1,BF
5750 PSET(60,100),1:COLOR11:PRINT#1,"Vuo
i continuare?":COLOR4:PSET(108,115),1:P
RINT#1,"(S/N)":B$=""
5760 B$=INKEY$:IF B$="" THEN 5760
5770 IF B$="S" THEN GOTO 5040 ELSE IF B$<
>"N" THEN 5760
5780 CC=0:DD=0
5790 FOR I=1 TO 40
5800 IF A$(I)="" THEN 5820 ELSE D=D+1:A2$
=MID$(A$(I),D,1):A3$=MID$(D1$(I),D,1)
5810 EE%=EE%+ASC(A2$):FF%=FF%+ASC(A3$):I
F D=K(I) AND EE=FF THEN CC=CC+1:D=0:EE%=0
:FF%=0:NEXT I:ELSE IF D<K(I) THEN GOTO 58
00
5820 NEXT I
5830 FOR I=1 TO 40
5840 IF A1$(I)="" THEN 5860 ELSE D=D+1:A2
$=MID$(A1$(I),D,1):A3$=MID$(D2$(I),D,1)
5850 EE%=EE%+ASC(A2$):FF%=FF%+ASC(A3$):I
F D=K1(I) AND EE=FF THEN DD=DD+1:D=0:EE%=
0:FF%=0:NEXT I:ELSE IF D<K1(I) THEN GOTO
5840
5860 NEXT I
5870 LINE(0,0)-(255,192),1,BF
5880 PSET(8,40),1:COLOR11:PRINT#1,"Hai i
ndovinato":COLOR4:PRINT#1,CC:PSET(8,55)
,1:COLOR11:PRINT#1,"definizioni orizzont
ali su":COLOR4:PRINT#1,AA
5890 PSET(8,80),1:COLOR9:PRINT#1,"e":CO
LOR2:PRINT#1,DD:COLOR9:PSET(8,95),1:PRIN
T#1,"definizioni verticali su":COLOR2:P
RINT#1,BB
5900 PSET(72,115),1:COLOR5:PRINT#1,"BATT
I UN TASTO":B$=""
5910 B$=INKEY$:IF B$="" THEN 5910 ELSE RE
TURN
6000 LINE(0,0)-(255,192),1,BF
6010 PSET(32,40),1:COLOR2:PRINT#1,"max d
ef orizzontali":AA
6020 PSET(32,70),1:COLOR6:PRINT#1,"SCEGL
I":B$="":NO$="":N$="":L=0:J=86:B=0
6030 NO$=INKEY$:IF NO$="" THEN 6030
6040 L=L+LEN(NO$):PSET(J,70),1:PRINT#1,N
O$
6050 B$=B$+NO$:B=VAL(B$):IF L<2 THEN 606
0 ELSE IF L=2 THEN 6080
6060 J=J+6:GOTO 6030
6070 LINE(86,70)-(255,80),1,BF:GOTO 6020
6080 N$=INKEY$:IF N$="" THEN 6080
6090 IF ASC(N$)=13 THEN 6100
6100 IF B<0 XOR B=0 XOR B>AA THEN 6070 E
LSE 6110
6110 LINE(0,0)-(255,192),1,BF
6120 PSET(8,10),1:COLOR9:PRINT#1,B;"ORIZ

```

Seguito listato Cruci Max.

```

ZONTALE":PSET(8,30),1:COLOR11:PRINT#1,D
0$(B):FOR I=1 TO 500:NEXT I
6130 PSET(8,100),1:COLOR4:PRINT#1,"Da ad
esso potrai scrivere quel- la che tu pen
si sia la defini- zione giusta."
6140 A$(B)="":B$="":L=0:J=8:X=140
6150 B$=INKEY$:IF B$=""THEN 6150
6160 L=L+LEN(B$):PSET(J,X),1:COLOR11:PRI
NT#1,B$
6170 A$(B)=A$(B)+B$:IF ASC(B$)=13 THEN R
ETURN
6180 J=J+6:IF J>256 THEN J=8:X=X+10:GOTO
6150 ELSE 6150
7000 LINE(0,0)-(255,192),1,BF
7010 PSET(52,40),1:COLOR10:PRINT#1,"max
de# verticali";BB
7020 PSET(52,70),1:COLOR6:PRINT#1,"SCEGL
I":B$="":N0$="":N$="":L=0:J=102:B=0
7030 N0$=INKEY$:IF N0$=""THEN 7030
7040 L=L+LEN(N0$):PSET(J,70),1:PRINT#1,N
0$
7050 B$=B$+N0$:B=VAL(B$):IF L<2 THEN 706
0 ELSE IF L=2 THEN 7080
7060 J=J+6:GOTO 7030
7070 LINE(102,70)-(255,80),1,BF:GOTO 702
0
7080 N$=INKEY$:IF N$=""THEN 7080
7090 IF ASC(N$)=13 THEN 7100
7100 IF B<0 XOR B=0 XOR B>BB THEN 7070 E
LSE 7110
7110 LINE(0,0)-(255,192),1,BF
7120 PSET(8,10),1:COLOR5:PRINT#1,B;"VERT
ICALE":PSET(8,30),1:COLOR11:PRINT#1,DV$
(B):FOR I=1 TO 500:NEXT I
7130 PSET(8,100),1:COLOR2:PRINT#1,"Da ad
esso potrai scrivere quel- la che tu pen
si sia la defini- zione giusta."
7140 A1$(B)="":B$="":L=0:J=8:X=140
7150 B$=INKEY$:IF B$=""THEN 7150
7160 L=L+LEN(B$):PSET(J,X),1:COLOR11:PRI
NT#1,B$
7170 A1$(B)=A1$(B)+B$:IF ASC(B$)=13 THEN
RETURN
7180 J=J+6:IF J>256 THEN J=8:X=X+10:GOTO
7150 ELSE 7150
10000 COLOR,14,14:SCREEN 2
10010 DRAW "BM45,96BU96":DRAW "C1610BU10
BR10R30F10D10610U10BL10BU10F10BU10BL10L1
0610D10F10R20F10D20610L40BU10BR10610BU10
BR10R20BU10BR10610BU10BR10U10L30BU10BL10
F10BL10BU10U30":PAINT(45,40),1,1
10020 DRAW "BM85,80":DRAW "C1U5R5D5L5":P
AINT(86,79),1,1
10030 DRAW "BM94,96BU96":DRAW "C1D10R10U
10L10BD20D60R10U60L10":PAINT(95,5),1,1:P
AINT(95,40),1,1
10040 DRAW "BM107,80":DRAW "C1U5R5D5L5":

```

```

PAINT(108,79),1,1
10050 DRAW "BM115,96BU96":DRAW "C1D80R40
BU10BR10G10BR10BU10L40U70L10":PAINT(117,
40),1,1
10060 DRAW "BM165,80":DRAW "C1U5R5D5L5":
PAINT(166,79),1,1
10070 DRAW "BM173,96BU96":DRAW "C1BR10G1
0D60F10R30BU10BR10G10BR10BU10L30BU10BL10
F10BL10BU10U40BU10BR10G10BR30BU10BR10L30
BR30BU10BL10F10BU10BL10L30":PAINT(174,40
),1,1
10080 DRAW "BM223,80":DRAW "C1U5R5D5L5":
PAINT(224,79),1,1
10090 LINE(38,82)-(98,99),2,BF:LINE(99,8
2)-(159,99),15,BF:LINE(160,82)-(221,99),
6,BF
10100 OPEN "GRP:" FOR OUTPUT AS #1
10110 FOR SA=1 TO 1000000#
10120 FOR SB=1 TO 15
10130 FOR SC=0 TO 1
10140 PSET(96,102):COLOR SB:PRINT #1,"PR
ESENTA"
10145 PSET(92,142),14:PRINT#1,"CRUCI/MSX
"
10150 PSET(72,184):PRINT #1,"Batti un ta
sto"
10160 A$=INKEY$:IF A$="" THEN 10170 ELSE
RETURN
10170 NEXT SC,SB,SA
10180 RETURN
15000 AA=31:BB=27:XX=AA+BB:RESTORE 15030
15010 FOR I=1 TO 31
15020 READ D0$(I)
15030 DATA Il Bongusto noto cantante,Le
lancette dell'orologio,Iniz.di Dumas,Un
prepotente signorotto,Il periodo dei veg
lioni
15040 DATA Le gode chi è raccomandato,Il
nome di Manfredi,Viene fissato sulla si
ringa,Fianco,La retribuzione basata sul
ren- dimento
15050 DATA Il principio di Eulero,Sono p
ari... in corsa,Abbandonò Didone,Il fil
o del pescatore,L'ultima nota
15060 DATA L'autore di Nanà,Colorati art
ificialmente,Grosse candele,L'autore de
La cittadella,Il cantore di Orlando
15070 DATA La si rende a Dio,Hanno scars
e cognizioni,Parma,Le vocali di molti,Un
o dei continenti
15080 DATA Non intossica chi ha la masch
e- ra,Una preposizione,Può diventare un
poker,S'infilano nelle tasche,Cose da nu
lla,Scorre nella Valcamonica
15090 NEXT I
15100 RESTORE 15130
15110 FOR I=1 TO 27
15120 READ DV$(I)
15130 DATA La Bertini del cinema auto,So

```


Seguito Ilistato Cruci Msx.

```

no diplomati, Non gravato da imposte, Un s
aluto arabo, Il frutto dell'orticaria
15140 DATA Il Biagi giornalista, Colpevol
i, Le compiono i ginnasti, Appanna lo spec
chio, Una particella nobiliare
15150 DATA Nel collo e nel cranio, La par
ità farmaceutica, Risultano sulla pagella
, Intestardirsi, Ingresso
15160 DATA Pesano sul groppone, Fare... v
ersi da cane, Una membrana dell'orecchio,
Il calcio da undici metri, Sondrio
15170 DATA Verdeggia nel deserto, L'amico
di Eurialo, Ritornano all'alveare, Allegr
i/briosi, Il centro di Vienna
15180 DATA A te, Metro lineare
15190 NEXT I
15200 RESTORE 15230
15210 FOR I=1 TO 31
15220 READ RV(I), RO(I), K(I), D1$(I)
15230 DATA 30, 26, 4, FRED, 110, 26, 5, SFERE, 2
06, 26, 2, AD, 30, 36, 3, RAS, 94, 36, 9, CARNEVALE
, 30, 46, 12, AGEVOLAZIONI
15240 DATA 30, 56, 4, NINO, 110, 56, 3, AGO, 174
, 56, 4, LATO, 30, 66, 7, COTTIMO, 158, 66, 2, EU
15250 DATA 206, 66, 2, OS, 30, 76, 4, ENEA, 126,
76, 5, LENZA, 30, 86, 2, SI, 78, 86, 4, ZOLA
15260 DATA 158, 86, 5, TINTI, 30, 96, 4, CERI, 1
42, 96, 6, CRONIN, 30, 106, 7, ARIOSTO, 158, 106,
5, ANIMA
15270 DATA 46, 116, 9, IGNORANTI, 206, 116, 2,
PR, 62, 126, 2, OE, 110, 126, 4, ASIA, 190, 126, 3,
GAS
15280 DATA 30, 136, 3, PER, 94, 136, 4, TRIS, 17
4, 136, 4, MANI, 30, 146, 6, INEZIE, 142, 146, 5, 0
GLIO
15290 NEXT I
15300 RESTORE 15330
15310 FOR I=1 TO 27
15320 READ R1(I), R2(I), K1(I), D2$(I)
15330 DATA 30, 26, 9, FRANCESCA, 46, 26, 10, RA
GIONIERI, 62, 26, 6, ESENTE, 110, 26, 5, SALAM, 1
26, 26, 7, FRAGOLA
15340 DATA 142, 26, 4, ENZO, 158, 26, 3, REI, 17
4, 26, 10, EVOLUZIONI, 206, 26, 5, ALITO, 222, 26
, 2, DE
15350 DATA 94, 36, 2, CO, 190, 36, 3, ANA, 78, 46
, 9, VOTAZIONI, 222, 56, 9, OSTINARSI, 158, 66, 7
, ENTRATA
15360 DATA 190, 76, 4, ANNI, 110, 86, 7, LATRAR
E, 206, 86, 7, TIMPANO, 62, 96, 6, RIGORE, 94, 106
, 2, SU
15370 DATA 126, 106, 4, OASI, 142, 116, 4, NISO
, 30, 126, 3, API, 190, 126, 3, GAI, 46, 136, 2, EN
15380 DATA 94, 136, 2, TI, 174, 136, 2, ML
15390 NEXT I
15400 RETURN
17000 AA=25:BB=28:XX=AA+BB:RESTORE 17030

```

```

17010 FOR I=1 TO 25
17020 READ D0$(I)
17030 DATA Uno sport, Sorveglianza un palazz
o, Velocissime siluranti, Le cascatelle de
i fiumi, Istituto Ricostruzione Indus-
triale
17040 DATA Liriche in versi, Si cimentano
in gare, E' il tipico innamorato, Gira a p
oppa, Era ora
17050 DATA L'argento, Lo punisce l'arbitr
o, Eccellenti, Conservano i testamenti, C'è
quella dello scandalo
17060 DATA Un mese freddo, Un bel voto, Il
popolare Pacino, Chiarire con argomenti
validi, La santa da Cascia
17070 DATA I sostegni dei ciocchi nel ca
- minetto, Quando piove permettono di ca
m- minare nelle vie anche a chi non h
a l'ombrello, Risonanza, Muore con Radames
, Il nome di Nazzari
17080 NEXT I
17090 RESTORE 17120
17100 FOR I=1 TO 28
17110 READ DV$(I)
17120 DATA Lo si dice porgendo, Il portaf
ortuna nel prato!, Scottate, Precede il Mi
ssa est, Sono simili ai ravioli
17130 DATA La Negri poetessa, L'ultima le
ttera, Una frittatina ripiena, Napoli, I ri
sultati finali
17140 DATA Oscuri, Sono uguali in teatro!
, Esegue pregevoli lavori d'ago, Partecipa
no al voto, Ridotti in cattivo stato
17150 DATA Il fiume di Bottego, Piacevoli
comodità, Così è la mosca che non se ne
va!, Cosparsa di spine, Gradini di legno
17160 DATA Il giaciglio con le corde, La
fine di Ivanhoe, Il più alto strumento a
corda, Quando è stipato non si può sa- li
rvi, Una preposizione
17170 DATA In mezzo al lido, Iniz della D
use, Le vocali nel conto
17180 NEXT I
17190 RESTORE 17220
17200 FOR I=1 TO 25
17210 READ RV(I), RO(I), K(I), D1$(I)
17220 DATA 30, 26, 11, EQUITAZIONE, 30, 36, 7,
CUSTODE, 158, 36, 3, MAS, 30, 46, 9, CATARATTE, 1
90, 46, 3, IRI
17230 DATA 30, 56, 3, ODI, 126, 56, 6, ATLETI, 4
6, 66, 5, ROMEO, 158, 66, 5, ELICA, 30, 76, 10, FIN
ALMENTE
17240 DATA 206, 76, 2, AG, 46, 86, 5, FALLO, 142
, 86, 6, OTTIMI, 30, 96, 5, NOTAI, 126, 96, 6, PIET
RA
17250 DATA 46, 106, 7, GENNAIO, 174, 106, 4, OT
TO, 30, 116, 2, AL, 78, 116, 10, DIMOSTRARE, 30, 1
26, 4, RITA
17260 DATA 110, 126, 5, ALARI, 30, 136, 7, PORT

```

Seguito listato Cruci Msx.

ICI,190,136,3,ECO,62,146,4,AIDA,142,146,
 6,AMEDEO
 17270 NEXT I
 17280 RESTORE 17310
 17290 FOR I=1 TO 28
 17300 READ R1(I),R2(I),K1(I),D2\$(I)
 17310 DATA 30,26,4,ECCO,46,26,12,QUADRIF
 OGILIO,62,26,9,USTIONATE,78,26,3,ITA,94,2
 6,10,TORTELLINI
 17320 DATA 110,26,3,ADA,126,26,4,ZETA,15
 8,26,8,OMELETTE,174,26,2,NA,190,26,5,ESI
 TI
 17330 DATA 222,26,3,BUI,142,46,2,TT,206,
 46,11,RICAMATRICE,174,56,8,ELETTORI,78,6
 6,9,MALANDATI
 17340 DATA 110,66,3,OMO,222,66,3,AGI,142
 ,76,6,NOIOSA,190,86,4,IRTE,126,96,5,PIOL
 I
 17350 DATA 110,106,5,AMACA,222,106,2,OE,
 30,116,4,ARPA,158,116,4,TRAM,62,126,3,TR
 A
 17360 DATA 94,136,2,ID,190,136,2,ED,222,
 136,2,00
 17370 NEXT I
 17380 RETURN
 19000 AA=30:BB=31:XX=AA+BB:RESTORE 19030
 19010 FOR I=1 TO 30
 19020 READ D0\$(I)
 19030 DATA Disaccordi di breve durata,Co
 pre il cassone del camion,Un opera in ve
 rsi,Le monete spagnole,Un canto patriott
 ico
 19040 DATA Il dente più aguzzo,Risonanza
 ,Un apertura nel pavimento,Però,Il giorn
 o più breve
 19050 DATA Scorgere,Il terzo figlio di A
 damo,Accogliere in casa,Una virtù teolog
 ale,La indossa il meccanico
 19060 DATA Li invoca l'esasperato,Si car
 ica sul basto,Il nome di Cimarosa,In mez
 zo,Quando decide di passare il Ru- bicon
 e disse*IL DADO è TRATTO*
 19070 DATA Istituto Regionale,Le vocali
 nel testo,Un animale da pelliccia,Si raf
 fronta all'...altra,Formano la vigna
 19080 DATA Il capoluogo della Stiria,Una
 distesa tra i monti,Li... spennano certi
 imbroglio- ni,L'ippodromo dei napoletan
 i,Un locale sotto il tetto
 19090 NEXT I
 19100 RESTORE 19130
 19110 FOR I=1 TO 31
 19120 READ DV\$(I)
 19130 DATA Lo usa il rosticciere,Un adun
 anza di cardinali,Bagna Bonn,Il sangue n
 ei prefissi,La fine della conferenza
 19140 DATA Un cantante come Pavarotti,Mi

ngherlino,La gran voglia di lavorare,Il
 gigante fratello di Efialte,Il sodio
 19150 DATA Una stagione,Un Sandro della
 TV,La agita il cane,Segue l'alfa greca,G
 iovano ai malati
 19160 DATA E'dura per chi stenta,Si gett
 ano nei solchi,Il leone d'America,Corda,
 Corvina
 19170 DATA La vasta prateria russa,Non b
 asta mai!,Sono tutti salati,Il cavallo a
 mico di Topolino,Un tratto dell'intestin
 o
 19180 DATA La Wertmuller regista,Le lanc
 ia chi è atterrito,Il... tifoso di una di
 va,Umilia il portiere,In mezzo al rigo,A
 ttraversa il Piemonte
 19190 NEXT I
 19200 RESTORE 19230
 19210 FOR I=1 TO 30
 19220 READ RV(I),RO(I),K(I),D1\$(I)
 19230 DATA 30,26,6,SCREZI,142,26,6,TELON
 E,30,36,5,POEMA,126,36,7,PESETAS,30,46,4
 ,INNO
 19240 DATA 110,46,6,CANINO,30,56,3,ECO,9
 4,56,6,BOTOLA,206,56,2,MA,30,66,2,DI
 19250 DATA 78,66,6,VEDERE,190,66,3,SET,3
 0,76,8,OSPITARE,174,76,4,FEDE,46,86,4,TU
 TA
 19260 DATA 158,86,4,NUMI,30,96,4,SOMA,11
 0,96,8,DOMENICO,30,106,3,TRA,94,106,6,CE
 SARE
 19270 DATA 206,106,2,IR,30,116,2,EO,78,1
 16,6,LONTRA,190,116,3,UNA,62,126,6,FILAR
 I
 19280 DATA 174,126,4,GRAZ,30,136,7,PIANO
 RO,158,136,5,POLLI,30,146,6,AGNANO,142,1
 46,6,SOLAIO
 19290 NEXT I
 19300 RESTORE 19330
 19310 FOR I=1 TO 31
 19320 READ R1(I),R2(I),K1(I),D2\$(I)
 19330 DATA 30,26,6,SPIEDO,46,26,10,CONCI
 STORO,62,26,4,RENO,78,26,3,EMO,94,26,2,Z
 A
 19340 DATA 142,26,6,TENORE,158,26,5,ESIL
 E,174,26,4,LENA,190,26,3,OTO,206,26,2,NA
 19350 DATA 222,26,6,ESTATE,126,36,11,PAT
 ERNOSTRO,110,46,4,CODA,94,56,4,BETA,206,
 56,10,MEDICINALI
 19360 DATA 78,66,4,VITA,190,66,4,SEMI,62
 ,76,4,PUMA,174,76,4,FUNE,158,86,4,NERA
 19370 DATA 30,96,6,STEPPA,110,96,6,DENAR
 O,142,96,4,MARI,222,96,6,ORAZIO,94,106,5
 ,COLON
 19380 DATA 78,116,4,LINA,190,116,4,URLA,
 62,126,3,FAN,174,126,3,GOL,46,136,2,I6,1
 58,136,2,PO
 19390 NEXT I
 19400 RETURN

I computer si trovano molto spesso a elaborare grandi quantità di dati di ogni genere e ogni volta che l'elaboratore ha a che fare con archivi o raccolte di dati, è indispensabile poter reperire nel minor tempo possibile tutte le informazioni desiderate. Per fare ciò è necessario che i dati siano in ordine o che si abbia la possibilità di ordinarli.

Con il termine sorting viene indicato il problema dell'ordinamento all'interno degli elaboratori elettronici. Si potrebbero fare migliaia di esempi sull'importanza di questa tema, ma ora vogliamo trattare alcuni algoritmi proposti per la soluzione del problema dell'ordinamento. Presenteremo alcune routine implementate sul PB-700, che però possono essere facilmente adattate a ogni altro personal, cercando di farvi capire i vari metodi esaminati; per facilità useremo esempi numerici, fermo restando che gli ordinamenti si fanno su ogni tipo di entità (stringhe, record, file).

Innanzitutto si deve distinguere tra internal ed external sorting, dove, nel primo caso, l'ordinamento avviene su array, in memoria centrale, nell'altro sui file (questa eventualità si verifica ogni volta che i dati da ordinare non possono essere contenuti nella memoria del computer); noi ci occuperemo solo dell'internal sorting anche se accenneremo ai metodi che risolvono l'external.

Un po' d'ordine fra i nostri dati

Alcuni algoritmi di sorting utili per ogni tipo di problema

di Luca Zaninello

Ordinare n elementi a_1, a_2, \dots, a_n vuol dire poter riscrivere gli stessi elementi $a_{k1}, a_{k2}, \dots, a_{kn}$ in modo tale che specificata una funzione d'ordinamento f siano verificate le relazioni: $f(a_{k1}) < f(a_{k2}) < \dots < f(a_{kn})$.

Questi metodi si possono dividere in tre categorie: ordinamenti per inserzione, per selezione, per scambio.

Le prime tecniche

Il primo che esaminiamo è lo Straight insertion sort che funziona nel modo seguente: ci sono n elementi (nel nostro esempio otto numeri) scritti a caso; si confrontano i primi due e si scrivono

nella sequenza corretta, quindi si passa al terzo e lo si inserisce nella giusta posizione con i due numeri precedenti e così si continua fino a raggiungere la fine della lista. Gli elementi ancora da collocare, cioè, vengono esaminati sequenzialmente e inseriti nel posto giusto tra quelli già ordinati. La procedura che traduce questo algoritmo è quella delle linee 200-270. Per ogni algoritmo esistono degli indici di valutazione basati sul numero di confronti da fare tra gli elementi e sul numero di spostamenti a cui tali elementi sono soggetti. Un confronto tra i vari metodi può essere fatto con la tabella 1. Esiste un altro metodo, discendente da quello appena

Tabella 1 - Uno schemino in cui vengono confrontate le prestazioni dei vari metodi d'ordinamento.

		Straight insertion	Straight selection	Straight exchange (Bubblesort)	Shakesort	Heapsort	Quicksort
Numero di confronti degli elementi	Min	$n-1$	$\frac{n^2-n}{2}$	$\frac{n^2-n}{2}$	$n-1$	Per i metodi avanzati non esistono formule analoghe né per i confronti né per gli spostamenti, anche se si può dire che in genere sono proporzionali a $n \log n$.	
	Med	$\frac{n^2+n-2}{4}$	$\frac{n^2-n}{2}$	$\frac{n^2-n}{2}$	$\frac{n^2-nK-n \log n}{2}$		
	Max	$\frac{n^2-n-2}{2}$	$\frac{n^2-n}{2}$	$\frac{n^2-n}{2}$	$\frac{n^2-n}{2}$		
Spostamenti degli elementi	Min	$2(n-1)$	$3(n-1)$	0	0		
	Med	$\frac{n^2-9n-10}{4}$	$n(\log n + 0,57)$	$\frac{3}{4}(n^2-n)$	$\frac{3}{4}(n^2-n)$		
	Max	$\frac{n^2+3n-4}{2}$	$\frac{n^2+12n-12}{4}$	$\frac{3}{2}(n^2-n)$	$\frac{3}{2}(n^2-n)$		

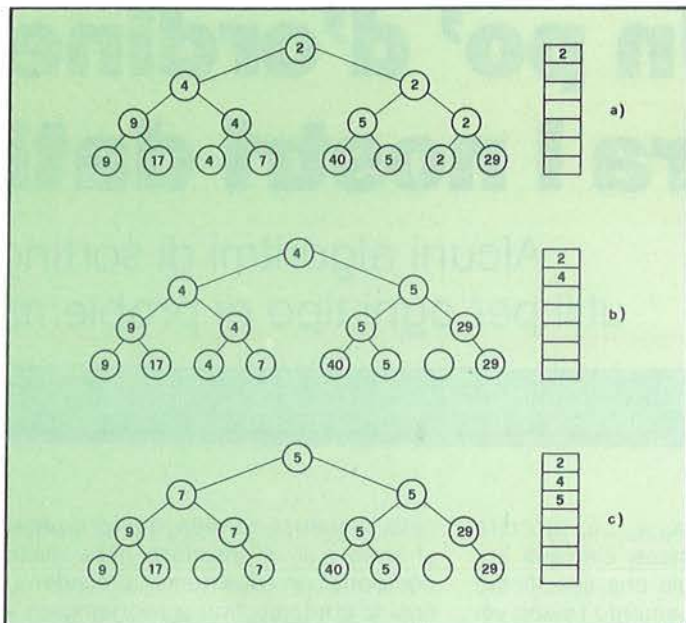


Figura 1 - Ecco come vengono selezionati gli elementi nello Heapsort: gli elementi sono otto, ma il numero di cerchi è 15 (cioè $2n-1$, $2 \cdot 8-1$). Qui vediamo le prime tre fasi.

presentato, che rende più veloce la ricerca del punto d'inserzione: l'ordinamento per inserzione binaria. L'idea nuova è quella di ricercare il punto di inserimento attraverso una bisezione continua della lista degli elementi. La procedura che realizza ciò è nelle linee 400-500. Rispetto all'inserzione diretta, si ha che i valori all'inizio vengono localizzati più in fretta, ma nel caso di sorting su elementi già ordinati, questo metodo è peggiore del precedente (ovviamente questo caso è abbastanza raro).

I metodi d'inserzione obbligano a degli shift (spostamenti) per fare posto col primo. Quindi si ripete quest'operazione per i restanti $n-1$ elementi fino a quando non si arriva all'ultimo confronto tra gli ultimi due numeri. La sequenza a cui si è arrivati è ordinata in modo crescente. Questo metodo è quasi opposto al primo il quale considera in ogni passata soltanto l'elemento successivo della sequenza sorgente e tutti quelli della sequenza di destinazione per trovare il punto d'inserzione: qui si considerano tutti i numeri della fila sorgente per trovare il minore e depositarlo nel primo posto disponibile della sequenza di destinazione.

La routine per questo algoritmo è alle righe 600-680.

Il quarto metodo che vediamo è caratterizzato dal confronto e scambio di due elementi; l'algoritmo viene chiamato Straight exchange sort, ma è meglio noto col nome di Bubblesort. Con questa denominazione si vuol dare l'idea che gli elementi più leggeri (di valore minore) salgono in superficie (ovvero s'avvicinano alla loro giusta posizione per andare in seguito a occuparla). La comprensione di quest'algoritmo è molto semplice e diciamo anche importante perché si presta a notevoli miglioramenti. Uno di questi è dato dal ricordare se si è avuto uno scambio durante una passata; ma questo algoritmo deve fare comunque $n-1$ passate (linea 800). Cerchiamo allora di ricordare la posizione dell'ultimo scambio; ma non è finita. Si può migliorare ancora guardando non solo in un verso della passata, ma anche nell'altro. Il frutto di tutti questi ragionamenti si concretizza nell'algoritmo di Shakersort che come potete facilmente notare non è altro che il Bubblesort applicato due volte (linee 1000-1100).

Qualcosa di più avanzato

Gli algoritmi appena visti non sono i più sofisticati, anche se con pochi ele-

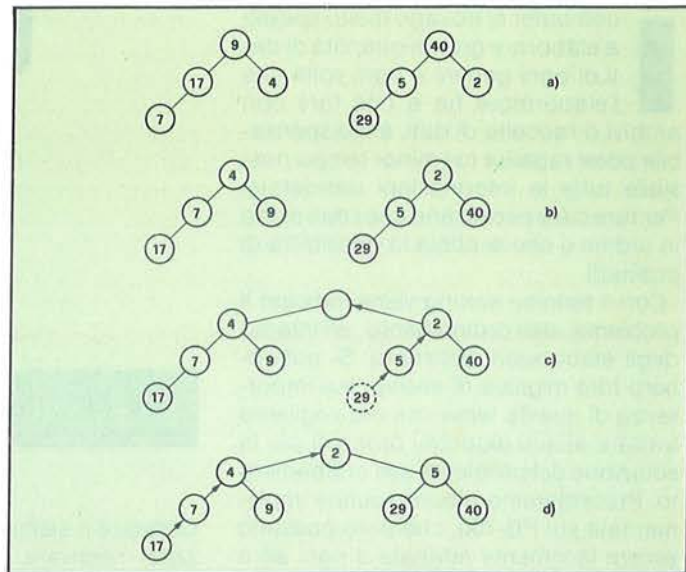


Figura 2 - Nella figura a) ci sono due heap ancora da ordinare; in b) sono stati ordinati (ogni figlio è maggiore di suo padre). In c) essi vengono congiunti e il numero minore migra verso il nodo radice, mentre tutti i suoi figli "crescono", avanzando di una posizione. E così si va avanti confrontando di volta in volta i due nodi sotto la radice, che sono maggiori dei rispettivi figli.

menti funzionano molto bene; passiamo allora all'esame di due modelli evoluti che, nel caso di grosse quantità di dati, sono i più veloci e utilizzati. Il primo metodo è l'Heapsort: esso nasce dal metodo di selezione del quale rappresenta un notevole miglioramento.

Anche in questo caso si tratta di tenere più informazioni possibili da ogni confronto tra elementi, non solo quindi riguardo all'ultimo. A partire dalla sequenza data, si crea una struttura a albero in cui ogni nodo è minore o uguale ai suoi nodi figli: è una specie di graduatoria parziale. Il nodo padre (la radice) conterrà l'elemento più piccolo di tutti: la fase successiva consisterà nell'eliminare tale elemento sostituendolo con il minore tra i rimasti, come mostrato nella figura 1.

La fase più complessa è quella iniziale in cui si fa in modo di usare un array di n elementi anziché uno di $2n-1$ elementi. Uno heap si può costituire con due sottoheap congiungendoli con un elemento che viene poi fatto migrare attraverso degli scambi verso l'elemento minore (figura 2). Da come è stato scritto il programma potrebbe sembrare che tale metodo non sia poi tanto veloce, ma in realtà non è così, sia perché l'istruzione di stampa occupa del tempo (e nel pro-

CASIO PB-100-200-300

gramma viene eseguita più spesso durante l'Heapsort che negli altri algoritmi), sia perché i metodi evoluti sono efficaci soprattutto con tanti elementi.

L'ultimo modello che presentiamo è il più veloce, come si capisce anche dal nome: il Quicksort.

Analizziamo il motivo di questa sua caratteristica: gli scambi fra elementi vanno effettuati tra lunghe distanze in modo da aver maggiore effetto (è quindi un miglioramento del Bubblesort). L'algoritmo funzionerà quindi così:

- si sceglie un elemento con indice K (il pivot);
- si divide l'array in due parti: $S = \{a[1], \dots, a[K-1]\}$ e $D = \{a[K+1], \dots, a[n]\}$ in modo tale che ogni elemento di S sia minore o uguale ad $a[K]$ e che ogni elemento di D sia maggiore ad $a[k]$;
- si ripetono i primi due passi applicandoli a S e D fino a quando si arriva a un

ordinamento totale.

La scelta del pivot influisce molto sul risultato finale; nel programma è stato scelto ogni volta nel punto medio della sequenza da esaminare ($x = \lfloor \text{int}((1+z)/2) \rfloor$). Anche in questo caso i risultati migliori si ottengono con molti elementi da ordinare.

Il problema dell'ordinamento esterno

Abbiamo parlato anche di external sorting che è la tecnica di ordinamento di dati residenti su file: essa differisce dall'internal sorting perché qui si può accedere, di solito, a un piccolo numero di elementi per volta. Inoltre ci sono dei vincoli legati alle operazioni di input/output da e verso file che bisogna minimizzare per rendere massima l'effi-

cienza; infine si può operare solo su elementi contigui. In questi casi la tecnica di ordinamento più diffusa è quella del merging: essa consiste nella combinazione di sequenze ordinate in un'unica sequenza attraverso delle scelte ripetute fra le componenti accessibili. Anche qui ci sono diversi algoritmi risolutivi: a esempio nello Straight merging, uno dei più semplici, si suddivide la sequenza in due metà, poi si fondono queste combinando elementi singoli in coppie e allineandoli in una nuova singola sequenza. Si prosegue, quindi, bisecando ancora la nuova sequenza ottenuta, combinando però le coppie in quadruple, e si va così avanti finché la sequenza non è del tutto ordinata. Oltre a questa tecnica ve ne sono altre come il Balanced merging, il Natural merging, il Multivay merging che però citiamo soltanto per completezza.

Figura 3 - Un esempio di ordinamento con meno di dieci numeri.

Figura 5 - Il numero massimo di elementi che il programma può ordinare (... e qui il Quicksort comincia a farla da padrone).

```

9 17 4 7 40 5 2 29 35
9 17 4 7 40 5 2 29 35
4 9 17 7 40 5 2 29 35
4 7 9 17 40 5 2 29 35
4 7 9 17 40 5 2 29 35
4 5 7 9 17 40 2 29 35
2 4 5 7 9 17 40 29 35
2 4 5 7 9 17 29 40 35
2 4 5 7 9 17 29 35 40
Straight insertion

66 62 43 34 13 32 41 25 6
6 66 62 43 34 13 32 41 25
6 13 66 62 43 34 25 32 41
6 13 25 66 62 43 34 32 41
6 13 25 32 66 62 43 34 41
6 13 25 32 34 66 62 43 41
6 13 25 32 34 41 66 62 43
6 13 25 32 34 41 43 66 62
6 13 25 32 34 41 43 62 66
Bubblesort
    
```

```

9 17 4 7 40 5 2 29 35 21 82 16 72 19 22 56 87 47 92 15 66 97 65 49 31
2 17 4 7 40 5 9 29 35 21 82 16 72 19 22 56 87 47 92 15 66 97 65 49 31
2 4 17 7 40 5 9 29 35 21 82 16 72 19 22 56 87 47 92 15 66 97 65 49 31
2 4 5 7 40 17 9 29 35 21 82 16 72 19 22 56 87 47 92 15 66 97 65 49 31
2 4 5 7 40 17 9 29 35 21 82 16 72 19 22 56 87 47 92 15 66 97 65 49 31
2 4 5 7 9 15 16 17 35 21 82 16 72 19 22 56 87 47 92 15 66 97 65 49 31
2 4 5 7 9 15 16 17 35 21 82 40 72 19 22 56 87 47 92 17 66 97 65 49 31
2 4 5 7 9 15 16 17 19 21 82 40 72 19 22 56 87 47 92 17 66 97 65 49 31
2 4 5 7 9 15 16 17 19 21 22 29 72 35 82 56 87 47 92 29 66 97 65 49 31
2 4 5 7 9 15 16 17 19 21 22 29 72 35 82 56 87 47 92 29 66 97 65 49 31
2 4 5 7 9 15 16 17 19 21 22 29 72 35 82 56 87 47 92 29 66 97 65 49 31
2 4 5 7 9 15 16 17 19 21 22 29 31 35 82 56 87 47 92 40 66 97 65 49 72
2 4 5 7 9 15 16 17 19 21 22 29 31 35 82 56 87 47 92 40 66 97 65 49 72
2 4 5 7 9 15 16 17 19 21 22 29 31 35 40 47 49 56 92 82 66 97 65 87 72
2 4 5 7 9 15 16 17 19 21 22 29 31 35 40 47 49 56 92 82 66 97 65 87 72
2 4 5 7 9 15 16 17 19 21 22 29 31 35 40 47 49 56 92 82 66 97 65 87 72
2 4 5 7 9 15 16 17 19 21 22 29 31 35 40 47 49 56 92 82 66 97 65 87 72
2 4 5 7 9 15 16 17 19 21 22 29 31 35 40 47 49 56 65 66 72 82 87 92 97
2 4 5 7 9 15 16 17 19 21 22 29 31 35 40 47 49 56 65 66 72 82 87 92 97
2 4 5 7 9 15 16 17 19 21 22 29 31 35 40 47 49 56 65 66 72 82 87 92 97
Straight selection

69 93 95 10 47 96 81 63 71 77 7 59 46 44 69 45 39 57 68 28 77 22 90 14 68
7 10 14 22 28 45 44 46 39 77 71 59 63 81 69 96 47 57 68 95 77 93 90 69 68
7 10 14 22 28 39 44 46 45 77 71 59 63 81 69 96 47 57 68 95 77 93 90 69 68
7 10 14 22 28 39 44 45 46 47 71 59 63 81 69 96 77 57 68 95 77 93 90 69 68
7 10 14 22 28 39 44 45 46 47 57 59 63 81 69 96 77 71 68 95 77 93 90 69 68
7 10 14 22 28 39 44 45 46 47 57 59 63 68 69 69 68 71 77 95 77 93 90 96 81
7 10 14 22 28 39 44 45 46 47 57 59 63 68 68 69 69 71 77 95 77 93 90 96 81
7 10 14 22 28 39 44 45 46 47 57 59 63 68 68 69 69 71 77 77 81 90 93 96 95
7 10 14 22 28 39 44 45 46 47 57 59 63 68 68 69 69 71 77 77 81 90 93 96 95
7 10 14 22 28 39 44 45 46 47 57 59 63 68 68 69 69 71 77 77 81 90 93 96 95
Quicksort
    
```

Figura 4 - Ecco altre tre tecniche di sorting.

```

82 70 86 91 39 77 14 52 56 38 44 79 45
70 82 86 91 39 77 14 52 56 38 44 79 45
70 82 86 91 39 77 14 52 56 38 44 79 45
70 82 86 91 39 77 14 52 56 38 44 79 45
39 70 82 86 91 77 14 52 56 38 44 79 45
39 70 77 82 86 91 14 52 56 38 44 79 45
14 39 70 77 82 86 91 52 56 38 44 79 45
14 39 52 70 77 82 86 91 56 38 44 79 45
14 39 52 56 70 77 82 86 91 38 44 79 45
14 38 39 52 56 70 77 82 86 91 44 79 45
14 38 39 44 52 56 70 77 82 86 91 79 45
14 38 39 44 52 56 70 77 79 82 86 91 45
14 38 39 44 45 52 56 70 77 79 82 86 91

```

Binary insertion

```

16 40 81 76 81 68 15 39 87 8 11 86 49
8 16 40 76 81 68 15 39 81 11 49 86 87
8 11 16 40 76 68 15 39 81 49 81 86 87
8 11 15 16 40 68 39 49 76 81 81 86 87
8 11 15 16 39 40 49 68 76 81 81 86 87
8 11 15 16 39 40 49 68 76 81 81 86 87

```

Shakersort

```

45 50 49 10 21 41 11 55 96 0 15 88 60
10 15 11 45 21 41 49 55 96 60 50 88 0
11 15 41 45 21 88 49 55 96 60 50 10 0
15 21 41 45 50 88 49 55 96 60 11 10 0
21 45 41 55 50 88 49 60 96 15 11 10 0
41 45 49 55 50 88 96 60 21 15 11 10 0
45 50 49 55 60 88 96 41 21 15 11 10 0
49 50 88 55 60 96 45 41 21 15 11 10 0
50 55 88 96 60 49 45 41 21 15 11 10 0
55 60 88 96 50 49 45 41 21 15 11 10 0
60 96 88 55 50 49 45 41 21 15 11 10 0
88 96 60 55 50 49 45 41 21 15 11 10 0
96 88 60 55 50 49 45 41 21 15 11 10 0
0 10 11 15 21 41 45 49 50 55 60 88 96

```

Heapsort

Com'è strutturato il programma

Abbiamo voluto illustrare le principali tecniche d'ordinamento e abbiamo pensato sia ai possessori del plotter che a quelli che ne sono privi. Cosicché se potete e volete usufruire della possibilità di stampare le varie fasi dell'ordinamento, vi verrà data anche la facoltà di scegliere il numero di elementi da ordinare. Alle linee 4000-4010 ci sono i primi numeri di partenza per essere ordinati: ciò è stato pensato sia per evitare all'utente di digitare fino a 25 numeri, sia perché i primi otto numeri riescono a essere scritti tutti in una linea del display del PB-700 (questo perde valore poi con l'uso del plotter).

Una volta che scegliete il modello di ordinamento da adoperare vi verrà prima visualizzato il suo nome sul video (linea 145), poi la sequenza di numeri di partenza. Ogni algoritmo ha bisogno di un certo numero di passate per arrivare all'ordinamento completo: ebbene il programma visualizza la situazione dopo ogni passata. Inoltre, dato che dopo tre passate i numeri sulla riga superiore iniziano a scomparire, si ha la possibilità di fermare lo scrolling semplicemente premendo un tasto (riga 2010). Da questo scoprirete una cosa interessante sui primi quattro metodi che non vi anticipiamo. Quando l'ordinamento è completato potete anche provare un altro metodo e tutto riprende da capo. Se non avete il plotter saranno ancora quegli otto numeri a girare (Gosub ST-200, quindi Gosub 1800, riga 185) altrimenti sarà il Casio a generarli a caso in un range da 0 a 99.

Per quanto riguarda la stampa su carta si fa anche uso dei colori (blu e rosso) per stampare la sequenza iniziale e il nome dell'algoritmo usato, al termine dell'ordinamento.

E con questo caso abbiamo proprio terminato; non vi resta che digitare il programma che speriamo possa aiutarvi nelle vostre applicazioni che hanno a che fare con ordinamenti di dati per scegliere l'algoritmo più adatto in un particolare problema. ■

CASIO PB-100-200-300

Listato 1 - Il programma in esame: le routine per ogni algoritmo sono ben evidenziate.

```

1 DATA1>Straight insertion,2>Binary
insertion,3>Straight selection
2 DATA4>Bubblesort,5>Shakersort,6>He
apsort,7>Quicksort
3 CLS :CLEAR :DIM M$(7)*20
4 FOR I=1 TO 7:READ M$(I):NEXT I
5 PRINT "Usi il plotter?":ST=2000:N
=8
10 Z$=INKEY$:IF Z$<>"N" THEN IF Z$<>"
S" THEN 10
15 IF Z$="N" THEN 45 ELSE LPRINT CHR$(
28):CHR$(37):DM=1:ST=3000
20 PRINT :PRINT "          < 8-25 >
";
25 LOCATE 0,1:INPUT "Quanti numeri vu
oi ";N
30 IF N<8 THEN N=8 ELSE IF N>25 THEN
N=25
35 IF N<10 THEN DM=2
37 IF N>13 THEN DM=0
40 LPRINT "S";DM
45 CLS :DRAW(0,0)-(159,0)-(159,31)-(0
,31)-(0,1)
50 DRAW(3,3)-(156,3)-(156,28)-(3,28)-
(3,4)
55 DRAW(1,1)-(3,3):DRAW(158,1)-(156,3
):DRAW(158,30)-(156,28):DRAW(1,30)-(3,28
)
60 LOCATE 5,1:PRINT "Algoritmi"
65 LOCATE 4,2:PRINT "di Sortings"
70 DIM A(N),B(N),C(N*2)
75 FOR I=1 TO N:READ A(I):NEXT I
80 BEEP :BEEP 1
85 IF INKEY$="" THEN 85
90 CLS :PRINT "Scegli quale metodo di
ordinamento vuoi"
95 PRINT "usare tra quelli cheti most
rero";CHR$(96):
100 IF INKEY$="" THEN 100
105 CLS :DRAW(154,31)-(159,31)-(159,26
):DRAW(159,30)-(150,26)
110 FOR I=1 TO 7
115 LOCATE 0,I-1-INT(I/5)*4:PRINT M$(I
);
120 IF I=4 THEN IF INKEY$="" THEN 120
ELSE CLS
125 NEXT I:LOCATE 0,3:PRINT "0>elenco
algoritmi";
130 Z$=INKEY$:C=ASC(Z$):IF C<48 THEN 1
30

```

```

135 IF C>55 THEN 130
140 C=C-48:IF C=0 THEN 105 ELSE CLS
145 PRINT M$(C):FOR I=0 TO 99:NEXT I:C
LS
150 GOSUB 2000
160 DRAW(0,0)-(0,7)-(159,7)-(159,0)
165 IF ST=3000 THEN LPRINT "J1":GOSUB
3000:LPRINT "J0"
170 GOSUB 200*C:IF ST=3000 THEN GOSUB
1900
175 BEEP 1:PRINT "Altro metodo?";
180 Z$=INKEY$:IF Z$="N" THEN END ELSE
IF Z$<>"S" THEN 180
185 BEEP :BEEP :GOSUB ST-200
190 GOTO 105
195 REM*****
196 REM
197 REM      Straight insertion
198 REM
199 REM*****
200 FOR I=2 TO N
210 X=A(I):A(0)=X:J=I-1
220 IF X<A(J) THEN A(J+1)=A(J):J=J-1:G
OTO 220
230 A(J+1)=X
240 GOSUB ST
250 NEXT I
260 RETURN
395 REM*****
396 REM
397 REM      Binary insertion
398 REM
399 REM*****
400 FOR I=2 TO N
410 X=A(I):L=1:R=I-1
420 IF L>R THEN 450
430 M=INT((L+R)/2)
440 IF X<A(M) THEN R=M-1 ELSE L=M+1
445 GOTO 420
450 FOR J=I-1 TO L STEP -1
460 A(J+1)=A(J):NEXT J
470 A(L)=X:GOSUB ST
480 NEXT I
490 RETURN
595 REM*****
596 REM
597 REM      Straight selection
598 REM
599 REM*****
600 FOR I=1 TO N-1
610 K=I:X=A(I)
620 FOR J=I+1 TO N
630 IF A(J)<X THEN K=J:X=A(J)

```

CASIO PB-100-200-300

Seguito listato 1

```

640 NEXT J
650 A(K)=A(I):A(I)=X
655 GOSUB ST
660 NEXT I
670 RETURN
795 REM*****
796 REM
797 REM    Bubblesort
798 REM
799 REM*****
800 FOR I=2 TO N
810 FOR J=N TO I STEP -1
820 IF A(J-1)>A(J) THEN X=A(J-1):A(J-1
)=A(J):A(J)=X
830 NEXT J
840 GOSUB ST
850 NEXT I
860 RETURN
995 REM*****
996 REM
997 REM    Shakersort
998 REM
999 REM*****
1000 L=2:R=N:K=N
1010 FOR I=R TO L STEP -1
1020 IF A(I-1)>A(I) THEN X=A(I-1):A(I-1
)=A(I):A(I)=X:K=I
1030 NEXT I
1040 L=K+1
1050 FOR I=L TO R
1060 IF A(I-1)>A(I) THEN X=A(I-1):A(I-1
)=A(I):A(I)=X:K=I
1070 NEXT I
1080 R=K-1:GOSUB ST
1090 IF L<=R THEN 1010
1100 RETURN
1195 REM*****
1196 REM
1197 REM    Heapsort
1198 REM
1199 REM*****
1200 L=INT(N/2)+1:R=N
1210 IF L>1 THEN L=L-1:GOSUB 1300:GOTO
1210
1220 IF R>1 THEN X=A(1):A(1)=A(R):A(R)=
X:R=R-1:GOSUB 1300:GOSUB ST:GOTO 1220
1230 FOR U=1 TO N:B(U)=A(N+1-U):NEXT U
1240 FOR U=1 TO N:A(U)=B(U):NEXT U
1250 GOSUB ST
1260 RETURN
1300 I=L:J=2*I:X=A(I)

```

```

1310 IF J>R THEN 1360
1320 IF J<R THEN IF A(J)>A(J+1) THEN J=
J+1
1330 IF A(I)<=A(J) THEN 1360
1340 X=A(I):A(I)=A(J):A(J)=X:I=J:J=2*I
1350 GOTO 1310
1360 RETURN
1395 REM*****
1396 REM
1397 REM    Quicksort
1398 REM
1399 REM*****
1400 S=1:B(1)=1:C(1)=N
1410 L=B(S):R=C(S):S=S-1
1420 I=L:J=R:X=A(INT((L+R)/2))
1430 IF A(I)<X THEN I=I+1:GOTO 1430
1440 IF A(J)>X THEN J=J-1:GOTO 1440
1450 IF I<=J THEN W=A(I):A(I)=A(J):A(J
)=W:I=I+1:J=J-1
1460 IF I<=J THEN 1430
1470 IF I<R THEN S=S+1:B(S)=I:C(S)=R
1480 R=J
1490 IF L<R THEN 1420
1500 GOSUB ST
1510 IF S<>0 THEN 1410
1520 RETURN
1800 K=N
1810 FOR I=1 TO N
1820 X=INT(RND*K)+1
1830 B(I)=A(X):A(X)=A(K)
1840 K=K-1
1850 NEXT I
1860 FOR I=1 TO N:A(I)=B(I):NEXT I
1870 RETURN
1900 LPRINT "J3":LPRINT "F":2-DM:LPRINT
"S2"
1910 LPRINT "P":RIGHT$(M$(C),LEN(M$(C)
-2)
1920 LPRINT "J0":LPRINT "S":DM
1930 LPRINT "H":LPRINT "F2"
2000 FOR U=1 TO N:PRINT A(U):NEXT U:PR
INT
2010 IF INKEY$<>" " THEN 2010 ELSE RETUR
N
2800 FOR I=1 TO N:A(I)=INT(RND*100):NEX
T I
2810 RETURN
3000 FOR U=1 TO N:LPRINT "P":A(U):NEXT
U:LPRINT "H":LPRINT "F1":RETURN
4000 DATA 9,17,4,7,40,5,2,29,35,21,82,16
,72
4010 DATA 19,22,56,87,47,92,15,66,97,65,
49,31

```


Tappeto mobile

Completiamo il programma
con la pubblicazione delle shape

Pubblichiamo la seconda parte di Tappeto mobile: dopo aver presentato il programma in Basic, riportiamo qui le shape per la visualizzazione dei caratteri.

I simboli che vengono così creati sono quelli che compariranno nei settori dello schermo usati per il gioco. Ricordiamo che il listato in Basic e le spiegazioni del gioco sono comparse nel numero 35 di **Personal Software**.

(Fine)
(La precedente puntata è apparsa sul n. 35)

di Ugo Rossini

Parte seconda

Listato 1 - Il programma Tappeto mobile.

4000-	19 01 34	ORA	\$3401,Y	404D-	2D 0D 0D	AND	\$0D0D	4098-	49 0D	EOR	#\$0D
4003-	00	BRK		4050-	2D 0D 18	AND	\$180D	409A-	4D 09 DB	EOR	DB09
4004-	89	???		4053-	DF	???		409D-	DB	???	
4005-	00	BRK		4054-	3B	???		409E-	FB	???	
4006-	DE 00 4B	DEC	\$4B00,X	4055-	FF	???		409F-	DB	???	
4007-	01 A1	ORA	(\$A1,X)	4056-	1B	???		40A0-	03	???	
4008-	01 12	ORA	(\$12,X)	4057-	07	???		40A1-	4B	PHA	
400D-	02	???		4058-	68	PLA		40A2-	2D 0D 0D	AND	\$0D0D
400E-	5E 02 D6	LSR	\$D602,X	4059-	49 49	EOR	##49	40A5-	2D 0D 18	AND	\$180D
4011-	02	???		405B-	49 21	EOR	##21	40A8-	DF	???	
4012-	4A	LSR		405D-	DF	???		40A9-	3B	???	
4013-	03	???		405E-	DB	???		40AA-	FF	???	
4014-	C0 03	CPY	##03	405F-	DE	???		40AB-	1B	???	
4016-	39 04 AD	AND	\$AD04,Y	4060-	DB	???		40AC-	07	???	
4017-	04	???		4061-	6C 49 49	JMP	(\$4949)	40AD-	6B	PLA	
401A-	23	???		4064-	49 E1	EOR	##E1	40AE-	49 49	EOR	##49
401B-	05 99	ORA	\$99	4066-	DF	???		40B0-	49 21	EOR	##21
401D-	05 2B	ORA	\$2B	4067-	DB	???		40B2-	DF	???	
401F-	06 89	ASL	\$89	4068-	DB	???		40B3-	DB	???	
4021-	06 EF	ASL	\$EF	4069-	3B	???		40B4-	DB	???	
4023-	06 53	ASL	\$53	406A-	0B	PHP		40B5-	DB	???	
4025-	07	???		406B-	4D 49 49	EOR	\$4949	40B6-	6C 49 49	JMP	(\$4949)
4026-	D1 07	CMP	(\$07),Y	406E-	29 DB	AND	##DB	40B9-	49 E1	EOR	##E1
4028-	39 0B 99	AND	\$990B,Y	4070-	DF	???		40BB-	DF	???	
402B-	0B	PHP		4071-	DB	???		40BC-	DF	???	
402C-	DB	CLD		4072-	1B	???		40BD-	FB	???	
402D-	0B	PHP		4073-	1F	???		40BE-	3B	???	
402E-	F6 0B	INC	\$0B,X	4074-	48	PHA		40BF-	4B	PHA	
4030-	40	RTI		4075-	69 49	ADC	##49	40C0-	6D 49 29	ADC	\$2949
4031-	40	RTI		4077-	69 01	ADC	##01	40C3-	0D DB FB	ORA	\$FBDB
4032-	00	BRK		4079-	DB	CLD		40C6-	DB	???	
4033-	00	BRK		407A-	1B	???		40C7-	FB	???	
4034-	DB	???		407B-	DF	???		40C8-	03	???	
4035-	DB	???		407C-	FB	???		40C9-	4B	PHA	
4036-	92	???		407D-	1B	???		40CA-	69 49	ADC	##49
4037-	92	???		407E-	48	PHA		40CC-	69 01	ADC	##01
4038-	52	???		407F-	49 0D	EOR	##0D	40CE-	DB	CLD	
4039-	49 29	EOR	##29	4081-	4D 09 DB	EOR	\$DB09	40CF-	FB	???	
403B-	6D 49 DB	ADC	\$DB49	4084-	DB	???		40D0-	DB	???	
403E-	DB	???		4085-	FB	???		40D1-	FB	???	
403F-	1F	???		4086-	DB	???		40D2-	03	???	
4040-	DF	???		4087-	03	???		40D3-	4B	PHA	
4041-	1B	???		4088-	00	BRK		40D4-	09 4D	ORA	##4D
4042-	48	PHA		4089-	DB	???		40D6-	69 09	ADC	##09
4043-	49 0D	EOR	##0D	408A-	DB	???		40D8-	DB	CLD	
4045-	4D 09 DB	EOR	\$DB09	408B-	92	???		40D9-	DB	???	
4048-	DB	???		408C-	92	???		40DA-	3F	???	
4049-	FB	???		408D-	52	???		40DB-	DF	???	
404A-	DB	???		408E-	49 29	EOR	##29	40DC-	1B	???	
404B-	03	???		4090-	6D 49 DB	ADC	\$DB49	40DD-	00	BRK	
404C-	4B	PHA		4093-	DB	???		40DE-	DB	???	
				4094-	1F	???		40DF-	DB	???	
				4095-	DF	???		40E0-	9B	???	
				4096-	1B	???		40E1-	92	???	
				4097-	4B	PHA		40E2-	92	???	

Seguito listato Tappeto mobile.

40E3-	4A	LSR		4145-	FF	???	41A2-	DB	???
40E4-	49 49	EOR	##49	4146-	DB	???	41A3-	9B	???
40E6-	4D 49 09	EOR	\$0949	4147-	3B	???	41A4-	92	???
40E9-	DB	CLD		4148-	FF	???	41A5-	92	???
40EA-	DB	???		4149-	03	???	41A6-	4A	LSR
40EB-	3B	???		414A-	00	BRK	41A7-	2D 4D 49	AND \$494D
40EC-	1F	???		414B-	DB	EOR	41AA-	49 49	EOR ##49
40ED-	FF	???		414C-	DB	???	41AC-	DB	CLD
40EE-	DB	???		414D-	93	???	41AD-	3F	???
40EF-	03	???		414E-	92	???	41AE-	DF	???
40F0-	4B	PHA		414F-	52	???	41AF-	DB	???
40F1-	49 4D	EOR	##4D	4150-	49 49	EOR	41B0-	3B	???
40F3-	49 4D	EOR	##4D	4152-	4D 49 01	EOR	41B1-	3F	???
40F5-	09 DB	ORA	##DB	4155-	DB	CLD	41B2-	3F	???
40F7-	1B	???		4156-	DB	???	41B3-	2B	PLP
40F8-	DF	???		4157-	FB	???	41B4-	2D 2D 6D	AND \$6D2D
40F9-	DB	???		4158-	DF	???	41B7-	49 2D	EOR ##2D
40FA-	1B	???		4159-	DB	???	41B9-	2D 05 3B	AND \$3B05
40FB-	DF	???		415A-	4B	PHA	41BC-	3F	???
40FC-	03	???		415B-	49 4D	EOR	41BD-	3F	???
40FD-	4B	PHA		415D-	69 49	ADC	41BE-	FF	???
40FE-	69 49	ADC	##49	415F-	DB	CLD	41BF-	1B	???
4100-	49 09	EOR	##09	4160-	1B	???	41C0-	3F	???
4102-	4D DB DF	EOR	\$DFDB	4161-	DF	???	41C1-	3F	???
4105-	DB	???		4162-	DB	???	41C2-	3F	???
4106-	DB	???		4163-	DF	???	41C3-	0C	???
4107-	DB	???		4164-	03	???	41C4-	2D 2D 6D	AND \$6D2D
4108-	1F	???		4165-	4B	PHA	41C7-	09 2D	ORA ##2D
4109-	0B	PHP		4166-	69 49	ADC	41C9-	2D 2D 1C	AND \$1C2D
410A-	4D 49 49	EOR	\$4949	4168-	49 4D	EOR	41CC-	3F	???
410D-	49 09	EOR	##09	416A-	DB	CLD	41CD-	3F	???
410F-	05 1B	ORA	\$1B	416B-	DF	???	41CE-	DF	???
4111-	DF	???		416C-	DB	???	41CF-	DB	???
4112-	DB	???		416D-	DB	???	41D0-	3F	???
4113-	DB	???		416E-	FB	???	41D1-	FF	???
4114-	DB	???		416F-	0B	PHP	41D2-	4B	PHA
4115-	1B	???		4170-	4D 49 49	EOR	41D3-	49 69	EOR ##69
4116-	07	???		4173-	49 05	EOR	41D5-	49 29	EOR ##29
4117-	6B	PLA		4175-	FB	SED	41D7-	6D DB 1B	ADC \$1BDB
4118-	49 49	EOR	##49	4176-	DB	???	41DA-	DF	???
411A-	49 49	EOR	##49	4177-	DB	???	41DB-	DB	???
411C-	09 FC	ORA	##FC	4178-	DB	???	41DC-	DF	???
411E-	DB	???		4179-	DB	???	41DD-	DB	???
411F-	DB	???		417A-	0C	???	41DE-	4B	PHA
4120-	DB	???		417B-	4D 49 49	EOR	41DF-	49 09	EOR ##09
4121-	DB	???		417E-	49 05	EOR	41E1-	4D 69 49	EOR \$4969
4122-	1B	???		4180-	DB	CLD	41E4-	DB	CLD
4123-	6C 49 49	JMP	(\$4949)	4181-	DF	???	41E5-	DB	???
4126-	49 49	EOR	##49	4182-	DB	???	41E6-	FB	???
4128-	09 FC	ORA	##FC	4183-	DB	???	41E7-	FB	???
412A-	DB	???		4184-	FB	???	41E8-	DB	???
412B-	DB	???		4185-	4B	PHA	41E9-	1B	???
412C-	DB	???		4186-	69 49	ADC	41EA-	4B	PHA
412D-	DB	???		4188-	49 4D	EOR	41EB-	49 49	EOR ##49
412E-	1B	???		418A-	DB	CLD	41ED-	0D 4D 49	ORA \$494D
412F-	0C	???		418B-	1B	???	41F0-	DB	CLD
4130-	4D 41 41	EOR	\$4141	418C-	DF	???	41F1-	3B	???
4133-	49 09	EOR	##09	418D-	DB	???	41F2-	FF	???
4135-	05 1B	ORA	\$1B	418E-	DF	???	41F3-	3F	???
4137-	DF	???		418F-	03	???	41F4-	FF	???
4138-	DB	???		4190-	4B	PHA	41F5-	DB	???
4139-	1F	???		4191-	49 4D	EOR	41F6-	03	???
413A-	DF	???		4193-	69 49	ADC	41F7-	4B	PHA
413B-	DB	???		4195-	DB	CLD	41F8-	09 6D	ORA ##6D
413C-	07	???		4196-	DB	???	41FA-	29 6D	AND ##6D
413D-	4B	PHA		4197-	FB	???	41FC-	09 0D	ORA ##0D
413E-	4D 69 09	EOR	\$0969	4198-	DF	???	41FE-	DB	CLD
4141-	4D 69 DB	EOR	\$DB69	4199-	DB	???	41FF-	FB	???
4144-	3B	???		419A-	4B	PHA	4200-	FB	???
				419B-	49 09	EOR	4201-	DF	???
				419D-	4D 49 01	EOR	4202-	1B	???
				41A0-	00	BRK	4203-	DF	???
				41A1-	DB	???	4204-	4B	PHA

APPLE II

Seguito listato Tappeto mobile.

4205-	4D 29 4D	EOR	##4D29	4262-	92	???	42CF-	48	PHA		
4208-	29 4D	AND	##4D	4263-	4A	LSR	42D0-	49 49	EOR	##49	
420A-	01 DB	ORA	(#DB,X)	4264-	2D 2D 2D	AND	##2D2D	42D2-	4D 49 09	EOR	##0949
420C-	DB	???		4267-	2D 2D 2D	AND	##2D2D	42D5-	00	BRK	
420D-	DB	???		426A-	0D 38 3F	ORA	##3F38	42D6-	DB	???	
420E-	DB	???		426D-	3F	???		42D7-	DB	???	
420F-	3F	???		426E-	3F	???		42D8-	93	???	
4210-	FF	???		426F-	3F	???		42D9-	92	???	
4211-	00	BRK		4270-	3F	???		42DA-	92	???	
4212-	DB	???		4271-	3F	???		42DB-	4A	LSR	
4213-	DB	???		4272-	3F	???		42DC-	49 09	EOR	##09
4214-	92	???		4273-	67	???		42DE-	6D 49 09	ADC	##0949
4215-	92	???		4274-	2D 2D 2D	AND	##2D2D	42E1-	DB	CLD	
4216-	2A	ROL		4277-	2D 2D 2D	AND	##2D2D	42E2-	DB	???	
4217-	2D 2D 2D	AND	##2D2D	427A-	2D 05 DB	AND	##D805	42E3-	FB	???	
421A-	2D 2D 25	AND	##252D	427D-	DB	???		42E4-	FB	???	
421D-	DF	???		427E-	FB	???		42E5-	DB	???	
421E-	DF	???		427F-	DF	???		42E6-	03	???	
421F-	DB	???		4280-	DB	???		42E7-	28	PLP	
4220-	DF	???		4281-	1B	???		42EB-	2D 2D 2D	AND	##2D2D
4221-	0C	???		4282-	08	PHP		42EB-	2D 2D 2D	AND	##2D2D
4222-	4D 4D 69	EOR	##694D	4283-	2D 2D 2D	AND	##2D2D	42EE-	2D 1C DF	AND	##DF1C
4225-	29 18	AND	##18	4286-	6D 2D 2D	ADC	##2D2D	42F1-	DB	???	
4227-	DF	???		4289-	0D DB FB	ORA	##FBDB	42F2-	DB	???	
4228-	FB	???		428C-	FB	???		42F3-	DB	???	
4229-	DF	???		428D-	DF	???		42F4-	3B	???	
422A-	3B	???		428E-	DB	???		42F5-	08	PHP	
422B-	48	PHA		428F-	1F	???		42F6-	4D 49 49	EOR	##4949
422C-	4D 69 09	EOR	##0969	4290-	48	PHA		42F9-	49 29	EOR	##29
422F-	0D DB DF	ORA	##DFDB	4291-	4D 49 0D	EOR	##0D49	42FB-	18	CLC	
4232-	42	???		4294-	4D 4D DB	EOR	##DB4D	42FC-	DF	???	
4233-	42	???		4297-	FB	???		42FD-	DF	???	
4234-	1F	???		4298-	FB	???		42FE-	FF	???	
4235-	48	PHA		4299-	DF	???		42FF-	FB	???	
4236-	69 49	ADC	##49	429A-	1B	???		4300-	3B	???	
4238-	69 01	ADC	##01	429B-	DF	???		4301-	08	PHP	
423A-	DB	CLD		429C-	48	PHA		4302-	4D 49 49	EOR	##4949
423B-	FB	???		429D-	69 49	ADC	##49	4305-	49 29	EOR	##29
423C-	DB	???		429F-	0D 0D 4D	ORA	##4D0D	4307-	DB	CLD	
423D-	FB	???		42A2-	01 DB	ORA	(#DB,X)	4308-	DF	???	
423E-	03	???		42A4-	1B	???		4309-	DB	???	
423F-	48	PHA		42A5-	1F	???		430A-	DB	???	
4240-	09 4D	ORA	##4D	42A6-	1F	???		430B-	1B	???	
4242-	69 09	ADC	##09	42A7-	DF	???		430C-	1F	???	
4244-	DB	CLD		42A8-	FB	???		430D-	48	PHA	
4245-	1B	???		42A9-	1B	???		430E-	4D 49 49	EOR	##4949
4246-	DF	???		42AA-	48	PHA		4311-	09 0D	ORA	##0D
4247-	FB	???		42AB-	09 4D	ORA	##4D	4313-	DB	CLD	
4248-	1B	???		42AD-	69 0D	ADC	##0D	4314-	DF	???	
4249-	48	PHA		42AF-	4D 01 DB	EOR	##DB01	4315-	DB	???	
424A-	49 0D	EOR	##0D	42B2-	DB	???		4316-	DB	???	
424C-	4D 09 DB	EOR	##DB09	42B3-	FF	???		4317-	1B	???	
424F-	DB	???		42B4-	DF	???		4318-	1F	???	
4250-	1F	???		42B5-	DF	???		4319-	48	PHA	
4251-	DF	???		42B6-	1B	???		431A-	4D 49 49	EOR	##4949
4252-	1B	???		42B7-	48	PHA		431D-	09 0D	ORA	##0D
4253-	48	PHA		42B8-	49 4D	EOR	##4D	431F-	DB	CLD	
4254-	49 69	EOR	##69	42BA-	0D 6D 49	ORA	##496D	4320-	DF	???	
4256-	49 01	EOR	##01	42BD-	DB	CLD		4321-	DB	???	
4258-	DB	CLD		42BE-	DB	???		4322-	DB	???	
4259-	DB	???		42BF-	FB	???		4323-	1B	???	
425A-	FB	???		42C0-	1F	???		4324-	1F	???	
425B-	DB	???		42C1-	DF	???		4325-	48	PHA	
425C-	03	???		42C2-	DB	???		4326-	4D 49 49	EOR	##4949
425D-	00	BRK		42C3-	48	PHA		4329-	09 0D	ORA	##0D
425E-	DB	???		42C4-	49 09	EOR	##09	432B-	DB	CLD	
425F-	DB	???		42C6-	6D 4D 49	ADC	##494D	432C-	DF	???	
4260-	9B	???		42C9-	DB	CLD		432D-	DB	???	
4261-	92	???		42CA-	DB	???		432E-	DB	???	
				42CB-	DB	???		432F-	1B	???	
				42CC-	DF	???		4330-	1F	???	
				42CD-	DB	???		4331-	48	PHA	
				42CE-	1B	???		4332-	69 43	ADC	##43

Seguito Ilistato Tappeto mobile.

4334-	43	???		4391-	DF	???		43FB-	6D 4D 09	ADC	#094D
4335-	69 01	ADC	#01	4392-	6C 69 69	JMF	(\$6969)	43FE-	2D 4D 0D	AND	#0D4D
4337-	DB	CLD		4395-	09 4D	ORA	#04D	4401-	25 FF	AND	\$FF
4338-	1B	???		4397-	4D FC 3B	EOR	\$3BFC	4403-	DB	???	
4339-	FF	???		439A-	FF	???		4404-	1B	???	
433A-	DB	???		439B-	DB	???		4405-	DF	???	
433B-	FF	???		439C-	3B	???		4406-	DB	???	
433C-	1B	???		439D-	FF	???		4407-	3B	???	
433D-	4B	PHA		439E-	23	???		4408-	2C 4D 4D	BIT	\$4D4D
433E-	49 29	EOR	#29	439F-	4D 49 49	EOR	\$4949	440B-	69 09	ADC	#09
4340-	2D 4D 49	AND	\$494D	43A2-	49 49	EOR	#049	440D-	4D 25 FF	EOR	\$FF25
4343-	DB	CLD		43A4-	E1 DF	SBC	(\$DF, X)	4410-	DB	???	
4344-	DB	???		43A6-	1B	???		4411-	1B	???	
4345-	1B	???		43A7-	DF	???		4412-	DF	???	
4346-	FF	???		43AB-	FB	???		4413-	DB	???	
4347-	DB	???		43A9-	DB	???		4414-	3B	???	
4348-	1B	???		43AA-	07	???		4415-	0C	???	
4349-	00	BRK		43AB-	08	PHP		4416-	6D 09 4D	ADC	\$4D09
434A-	DB	???		43AC-	4D 69 2D	EOR	\$2D69	4419-	69 09	ADC	#09
434B-	DB	???		43AF-	0D 4D 29	ORA	\$294D	441B-	2D 18 FF	AND	\$FF18
434C-	9B	???		43B2-	DB	CLD		441E-	DB	???	
434D-	92	???		43B3-	DF	???		441F-	FB	???	
434E-	92	???		43B4-	DF	???		4420-	DB	???	
434F-	4A	LSR		43B5-	DB	???		4421-	3B	???	
4350-	49 29	EOR	#29	43B6-	DF	???		4422-	07	???	
4352-	2D 6D 49	AND	\$496D	43B7-	1F	???		4423-	4B	PHA	
4355-	01 DB	ORA	(\$DB, X)	43B8-	4B	PHA		4424-	6D 49 49	ADC	\$4949
4357-	1B	???		43B9-	29 4D	AND	#04D	4427-	09 6D	ORA	#06D
4358-	FF	???		43BB-	49 09	EOR	#09	4429-	DB	CLD	
4359-	DB	???		43BD-	6D 01 00	ADC	\$0001	442A-	3B	???	
435A-	3B	???		43C0-	DB	???		442B-	3F	???	
435B-	DF	???		43C1-	DB	???		442C-	3F	???	
435C-	03	???		43C2-	9B	???		442D-	3F	???	
435D-	4B	PHA		43C3-	92	???		442E-	3F	???	
435E-	69 49	ADC	#049	43C4-	92	???		442F-	FF	???	
4360-	49 09	EOR	#09	43C5-	4A	LSR		4430-	03	???	
4362-	4D DB DF	EOR	\$DFDB	43C6-	49 2D	EOR	#02D	4431-	4B	PHA	
4365-	DB	???		43C8-	2D 2D 4D	AND	\$4D2D	4432-	49 2D	EOR	#02D
4366-	DB	???		43CB-	09 DB	ORA	#0DB	4434-	44	???	
4367-	DB	???		43CD-	3B	???		4435-	44	???	
4368-	1F	???		43CE-	3F	???		4436-	4D 09 00	EOR	\$0009
4369-	0B	PHP		43CF-	3F	???		4439-	DB	???	
436A-	4D 49 69	EOR	\$6949	43D0-	3F	???		443A-	DB	???	
436D-	49 09	EOR	#09	43D1-	3F	???		443B-	9B	???	
436F-	05 1B	ORA	#1B	43D2-	FF	???		443C-	92	???	
4371-	DF	???		43D3-	03	???		443D-	92	???	
4372-	DB	???		43D4-	4B	PHA		443E-	4A	LSR	
4373-	3F	???		43D5-	6D 49 49	ADC	\$4949	443F-	49 2D	EOR	#02D
4374-	DF	???		43D8-	09 6D	ORA	#06D	4441-	2D 2D 4D	AND	\$4D2D
4375-	DB	???		43DA-	1B	CLC		4444-	09 DB	ORA	#0DB
4376-	07	???		43DB-	FF	???		4446-	3B	???	
4377-	6B	PLA		43DC-	DB	???		4447-	DF	???	
4378-	49 49	EOR	#049	43DD-	FB	???		4448-	DB	???	
437A-	49 49	EOR	#049	43DE-	DB	???		4449-	1B	???	
437C-	09 FC	ORA	#0FC	43DF-	3B	???		444A-	FF	???	
437E-	DB	???		43E0-	07	???		444B-	03	???	
437F-	3F	???		43E1-	0B	PHP		444C-	4B	PHA	
4380-	1F	???		43E2-	6D 09 4D	ADC	\$4D09	444D-	4D 49 49	EOR	\$4949
4381-	3F	???		43E5-	69 09	ADC	#09	4450-	49 0D	EOR	#0D
4382-	DF	???		43E7-	2D 3B DF	AND	\$DF3B	4452-	1B	CLC	
4383-	1B	???		43EA-	DB	???		4453-	DF	???	
4384-	6C 09 4D	JMP	(\$4D09)	43EB-	DB	???		4454-	DB	???	
4387-	0D 4D 4D	ORA	\$4D4D	43EC-	DB	???		4455-	3F	???	
438A-	21 DF	AND	(\$DF, X)	43ED-	1B	???		4456-	DF	???	
438C-	1F	???		43EE-	27	???		4457-	DB	???	
438D-	1F	???		43EF-	6D 69 49	ADC	\$4969	4458-	07	???	
438E-	1F	???		43F2-	49 4D	EOR	#04D	4459-	0B	PHP	
438F-	1F	???		43F4-	25 FF	AND	\$FF	445A-	4D 29 4D	EOR	\$4D29
4390-	1F	???		43F6-	DB	???		445D-	29 4D	AND	#04D
				43F7-	DB	???		445F-	29 FB	AND	#0FB
				43F8-	DB	???		4461-	1B	???	
				43F9-	DB	???		4462-	DF	???	
				43FA-	27	???		4463-	DB	???	

Seguito listato Tappeto mobile.

4464-	1B	???		44C8-	3F	???		452E-	01 DB	ORA	(#DB, X)
4465-	DF	???		44C9-	3F	???		4530-	DB	???	
4466-	23	???		44CA-	3F	???		4531-	DF	???	
4467-	4D 69 49	EOR	#4969	44CB-	1F	???		4532-	1B	???	
446A-	49 4D	EOR	#44D	44CC-	3F	???		4533-	DF	???	
446C-	21 DF	AND	(#DF, X)	44CD-	3F	???		4534-	DB	???	
446E-	DF	???		44CE-	3F	???		4535-	45 45	EOR	#45
446F-	DB	???		44CF-	27	???		4537-	29 2D	AND	#2L
4470-	DB	???		44D0-	4D 49 49	EOR	#4949	4539-	2D 6D 4D	AND	#4D6D
4471-	FB	???		44D3-	49 49	EOR	#449	453C-	49 DB	EOR	#4DB
4472-	23	???		44D5-	E1 DF	SBC	(#DF, X)	453E-	DB	???	
4473-	4D 4D 49	EOR	#494D	44D7-	DB	???		453F-	3B	???	
4476-	49 69	EOR	#469	44D8-	DB	???		4540-	DF	???	
4478-	21 DF	AND	(#DF, X)	44D9-	DB	???		4541-	1B	???	
447A-	DB	???		44DA-	1B	???		4542-	DF	???	
447B-	DB	???		44DB-	07	???		4543-	3F	???	
447C-	DB	???		44DC-	48	PHA		4544-	68	PLA	
447D-	DB	???		44DD-	4D 49 49	EOR	#4949	4545-	49 69	EOR	#469
447E-	23	???		44E0-	49 0D	EOR	#40D	4547-	49 69	EOR	#469
447F-	4D 49 4D	EOR	#4D49	44E2-	DB	CLD		4549-	49 3C	EOR	#43C
4482-	69 49	ADC	#449	44E3-	FB	???		454B-	DF	???	
4484-	21 DF	AND	(#DF, X)	44E4-	DB	???		454C-	DF	???	
4486-	DB	???		44E5-	DB	???		454D-	DB	???	
4487-	DF	???		44E6-	1B	???		454E-	FB	???	
4488-	FB	???		44E7-	DF	???		454F-	1B	???	
4489-	DB	???		44E8-	48	PHA		4550-	3F	???	
448A-	63	???		44E9-	2D 4D 49	AND	#494D	4551-	4C 69 09	JMP	#0969
448B-	4D 09 4D	EOR	#4D09	44EC-	09 2D	ORA	#2D	4554-	4D 49 29	EOR	#2949
448E-	69 49	ADC	#449	44EE-	0D DB FB	ORA	#FBDB	4557-	0D FC DF	ORA	#DFFC
4490-	05 18	ORA	#18	44F1-	DB	???		455A-	DB	???	
4492-	DF	???		44F2-	DB	???		455B-	1B	???	
4493-	DB	???		44F3-	1B	???		455C-	DF	???	
4494-	DB	???		44F4-	DF	???		455D-	1B	???	
4495-	DB	???		44F5-	48	PHA		455E-	3F	???	
4496-	1B	???		44F6-	09 4D	ORA	#4D	455F-	6C 49 09	JMP	(#0949)
4497-	07	???		44F8-	49 09	EOR	#409	4562-	4D 2D 2D	EOR	#2D2D
4498-	48	PHA		44FA-	4D 01 DB	EOR	#DB01	4565-	6D 3C DF	ADC	#DF3C
4499-	4D 49 49	EOR	#4949	44FD-	DB	???		4568-	3F	???	
449C-	49 0D	EOR	#40D	44FE-	DF	???		4569-	3F	???	
449E-	DB	CLD		44FF-	DB	???		456A-	3F	???	
449F-	3B	???		4500-	DF	???		456B-	FF	???	
44A0-	DF	???		4501-	1B	???		456C-	DF	???	
44A1-	DB	???		4502-	48	PHA		456D-	07	???	
44A2-	1B	???		4503-	09 2D	ORA	#2D	456E-	48	PHA	
44A3-	FF	???		4505-	4D 29 6D	EOR	#6D29	456F-	4D 29 2D	EOR	#2D29
44A4-	03	???		4508-	09 DB	ORA	#4DB	4572-	2D 6D 49	AND	#496D
44A5-	48	PHA		450A-	DB	???		4575-	DC	???	
44A6-	49 2D	EOR	#42D	450B-	DF	???		4576-	DB	???	
44A8-	2D 2D 4D	AND	#4D2D	450C-	DB	???		4577-	3B	???	
44AB-	09 00	ORA	#400	450D-	DF	???		4578-	3F	???	
44AD-	DB	???		450E-	1B	???		4579-	3F	???	
44AE-	DB	???		450F-	48	PHA		457A-	FF	???	
44AF-	9B	???		4510-	49 69	EOR	#469	457B-	DF	???	
44B0-	92	???		4512-	09 4D	ORA	#4D	457C-	48	PHA	
44B1-	92	???		4514-	49 DB	EOR	#4DB	457D-	09 2D	ORA	#2D
44B2-	4A	LSR		4516-	DB	???		457F-	2D 2D 4D	AND	#4D2D
44B3-	49 09	EOR	#409	4517-	1B	???		4582-	49 01	EOR	#401
44B5-	2D 4D 49	AND	#494D	4518-	1F	???		4584-	DB	CLD	
44B8-	01 DB	ORA	(#DB, X)	4519-	DF	???		4585-	DB	???	
44BA-	DB	???		451A-	DB	???		4586-	1B	???	
44BB-	1B	???		451B-	03	???		4587-	1F	???	
44BC-	1F	???		451C-	48	PHA		4588-	3F	???	
44BD-	DF	???		451D-	49 49	EOR	#449	4589-	FF	???	
44BE-	DB	???		451F-	4D 49 09	EOR	#0949	458A-	DB	???	
44BF-	03	???		4522-	00	BRK		458B-	48	PHA	
44C0-	48	PHA		4523-	DB	???		458C-	49 69	EOR	#469
44C1-	49 09	EOR	#409	4524-	DB	???		458E-	49 4D	EOR	#44D
44C3-	0D 4D 49	ORA	#494D	4525-	9B	???		4590-	49 DB	EOR	#4DB
44C6-	01 38	ORA	(#38, X)	4526-	92	???		4592-	DB	???	
				4527-	92	???		4593-	3F	???	
				4528-	4A	LSR		4594-	3F	???	
				4529-	49 49	EOR	#449	4595-	FF	???	
				452B-	2D 6D 49	AND	#496D	4596-	DB	???	

APPLE II

Seguito listato Tappeto mobile.

46C4-	69 49	ADC	##49	4725-	1B	???	4788-	DF	???		
46C6-	69 01	ADC	##01	4726-	08	PHP	4789-	23	???		
46C8-	DB	CLD		4727-	4D 29 6D	EOR	##6D29	478A-	6D 2D 0D	ADC	##0D2D
46C9-	DF	???		472A-	49 DB	EOR	##DB	478D-	2D 0D 2D	AND	##2D0D
46CA-	DB	???		472C-	DB	???		4790-	0D 25 1F	ORA	##1F25
46CB-	1B	???		472D-	DB	???		4793-	1F	???	
46CC-	1F	???		472E-	DB	???		4794-	1F	???	
46CD-	48	PHA		472F-	03	???		4795-	1F	???	
46CE-	4D 49 09	EOR	##0949	4730-	48	PHA		4796-	1F	???	
46D1-	0D 18 DF	ORA	##DF18	4731-	09 2D	ORA	##2D	4797-	1F	???	
46D4-	DB	???		4733-	6D 09 05	ADC	##0509	4798-	1F	???	
46D5-	DB	???		4736-	DB	CLD		4799-	1F	???	
46D6-	3B	???		4737-	47	???		479A-	6C 09 4D	JMP	##(4D09)
46D7-	08	PHP		4738-	47	???		479D-	69 09	ADC	##09
46D8-	4D 49 49	EOR	##4949	4739-	1F	???		479F-	4D E1 DF	EOR	##DFE1
46DB-	29 18	AND	##18	473A-	FF	???		47A2-	DF	???	
46DD-	DF	???		473B-	63	???		47A3-	FB	???	
46DE-	DF	???		473C-	09 0D	ORA	##0D	47A4-	1B	???	
46DF-	FB	???		473E-	6D 6D 01	ADC	##016D	47A5-	DF	???	
46E0-	3B	???		4741-	F8	SED		47A6-	07	???	
46E1-	08	PHP		4742-	3B	???		47A7-	08	PHP	
46E2-	0D 0D 0D	ORA	##0D0D	4743-	3F	???		47AB-	4D 4D 69	EOR	##694D
46E5-	0D 0D 05	ORA	##050D	4744-	1F	???		47AB-	09 4D	ORA	##4D
46E8-	18	CLC		4745-	DF	???		47AD-	05 18	ORA	##18
46E9-	FF	???		4746-	03	???		47AF-	DF	???	
46EA-	1B	???		4747-	08	PHP		47B0-	FB	???	
46EB-	DF	???		4748-	4D 29 0D	EOR	##0D29	47B1-	FB	???	
46EC-	3B	???		474B-	6D 01 DB	ADC	##DB01	47B2-	FB	???	
46ED-	07	???		474E-	1B	???		47B3-	1B	???	
46EE-	00	BRK		474F-	1F	???		47B4-	07	???	
46EF-	DB	???		4750-	DF	???		47B5-	48	PHA	
46F0-	DB	???		4751-	DB	???		47B6-	4D 4D 4D	EOR	##4D4D
46F1-	92	???		4752-	00	BRK		47B9-	4D 0D DB	EOR	##DB0D
46F2-	92	???		4753-	DB	???		47BC-	3B	???	
46F3-	92	???		4754-	DB	???		47BD-	1F	???	
46F4-	49 09	EOR	##09	4755-	9B	???		47BE-	1F	???	
46F6-	2D 4D 09	AND	##094D	4756-	92	???		47BF-	1F	???	
46F9-	DB	CLD		4757-	92	???		47C0-	1F	???	
46FA-	DB	???		4758-	52	???		47C1-	FF	???	
46FB-	1F	???		4759-	49 29	EOR	##29	47C2-	03	???	
46FC-	DF	???		475B-	6D 49 49	ADC	##4949	47C3-	48	PHA	
46FD-	1B	???		475E-	01 DB	ORA	##(DB,X)	47C4-	49 2D	EOR	##2D
46FE-	48	PHA		4760-	DB	???		47C6-	2D 2D 4D	AND	##4D2D
46FF-	49 0D	EOR	##0D	4761-	DB	???		47C9-	09 DB	ORA	##DB
4701-	4D 09 DB	EOR	##DB09	4762-	DF	???		47CB-	DB	???	
4704-	DB	???		4763-	FB	???		47CC-	DB	???	
4705-	1F	???		4764-	1B	???		47CD-	DF	???	
4706-	DF	???		4765-	48	PHA		47CE-	DB	???	
4707-	1B	???		4766-	49 49	EOR	##49	47CF-	1B	???	
4708-	48	PHA		4768-	4D 49 09	EOR	##0949	47D0-	00	BRK	
4709-	49 0D	EOR	##0D	476B-	DB	CLD		47D1-	DB	???	
470B-	4D 09 DB	EOR	##DB09	476C-	DB	???		47D2-	DB	???	
470E-	DB	???		476D-	DB	???		47D3-	93	???	
470F-	1F	???		476E-	DF	???		47D4-	92	???	
4710-	DF	???		476F-	DB	???		47D5-	92	???	
4711-	1B	???		4770-	1B	???		47D6-	09 2D	ORA	##2D
4712-	48	PHA		4771-	48	PHA		47D8-	2D 2D 2D	AND	##2D2D
4713-	49 0D	EOR	##0D	4772-	49 49	EOR	##49	47DB-	2D 0D DB	AND	##DB0D
4715-	4D 09 DB	EOR	##DB09	4774-	4D 49 09	EOR	##0949	47DE-	1F	???	
4718-	DB	???		4777-	DB	CLD		47DF-	3F	???	
4719-	1F	???		4778-	DB	???		47E0-	DF	???	
471A-	DF	???		4779-	DB	???		47E1-	DB	???	
471B-	1B	???		477A-	DF	???		47E2-	1F	???	
471C-	48	PHA		477B-	DB	???		47E3-	48	PHA	
471D-	49 0D	EOR	##0D	477C-	1B	???		47E4-	4D 49 2D	EOR	##2D49
471F-	4D 09 DB	EOR	##DB09	477D-	48	PHA		47E7-	0D 0D DB	ORA	##DB0D
4722-	DB	???		477E-	49 49	EOR	##49	47EA-	1F	???	
4723-	1F	???		4780-	4D 49 09	EOR	##0949	47EB-	3F	???	
4724-	DF	???		4783-	F8	SED		47EC-	DF	???	
				4784-	1B	???		47ED-	DB	???	
				4785-	DF	???		47EE-	1F	???	
				4786-	FB	???		47EF-	48	PHA	
				4787-	1B	???		47F0-	4D 49 49	EOR	##4949

Seguito listato Tappeto mobile.

47F3-	69	DB	ADC	##\$DB	484F-	DB	???	48B0-	29	DB	AND	##\$DB					
47F5-		DF	???		4850-	3B	???	48B2-		DF	???						
47F6-		DB	???		4851-	3F	???	48B3-		DB	???						
47F7-		3B	???		4852-	3F	???	48B4-		1B	???						
47F8-		1F	???		4853-	68	FLA	48B5-	6C	49	49	JMP (\$4949)					
47F9-		1F	???		4854-	49	69	EOR	##\$69	48B8-	0D	DB	DF	ORA	##\$DFDB		
47FA-		4B	PHA		4856-	49	29	EOR	##\$29	48BB-	DB	???					
47FB-	0D	6D	49	ORA	##\$496D	4858-	FB	SED		48BC-	3B	???					
47FE-	09	0D		ORA	##\$0D	4859-	DB	???		48BD-	0B	PHF					
4800-		DB	CLD		485A-	FB	???	48BE-	4D	49	09	EOR	##\$0949				
4801-		DF	???		485B-	DB	???	48C1-	05	FB	ORA	##\$FB					
4802-		DB	???		485C-	1B	???	48C3-	FB	???	???						
4803-		3B	???		485D-	0C	???	48C4-	DF	???	???						
4804-		1F	???		485E-	4D	49	69	EOR	##\$6949	48C5-	1F	???				
4805-		1F	???		4861-	49	FC	EOR	##\$FC	48C6-	4B	PHA					
4806-		4B	PHA		4863-	1B	???	48C7-	29	0D	AND	##\$0D					
4807-	4D	49	49	EOR	##\$4949	4864-	DF	???	48C9-	0D	2D	DB	ORA	##\$DB2D			
480A-	69	3B	ADC	##\$3B	4865-	DB	???	48CC-	FB	???	???						
480C-		3F	???		4866-	FB	???	48CD-	DF	???	???						
480D-		3F	???		4867-	4B	PHA	48CE-	1B	???	???						
480E-		3F	???		4868-	69	49	ADC	##\$49	48CF-	4B	PHA					
480F-		3F	???		486A-	09	4D	ORA	##\$4D	48D0-	49	4D	EOR	##\$4D			
4810-		3F	???		486C-	FC	???	48D2-	0D	DB	3B	ORA	##\$3BDB				
4811-		3F	???		486D-	FB	???	48D5-	DF	???	???						
4812-		67	???		486E-	DB	???	48D6-	DB	???	???						
4813-	4D	49	49	EOR	##\$4949	486F-	FB	???	48D7-	00	BRK						
4816-	49	05	EOR	##\$05	4870-	1B	???	48D8-	DB	???	???						
4818-		DB	CLD		4871-	4B	PHA	48D9-	92	???	???						
4819-		DF	???		4872-	69	49	ADC	##\$49	48DA-	4A	LSR					
481A-		DB	???		4874-	09	4D	ORA	##\$4D	48DB-	2D	0D	DB	AND	##\$DB0D		
481B-		DB	???		4876-	FC	???	48DE-	1F	???	???						
481C-		FB	???		4877-	1B	???	48DF-	1F	???	???						
481D-		4B	PHA		4878-	DF	???	48E0-	4B	PHA							
481E-	69	49	ADC	##\$49	4879-	DB	???	48E1-	0D	0D	DB	ORA	##\$DB0D				
4820-	49	6D	EOR	##\$6D	487A-	FB	???	48E4-	1F	???	???						
4822-		DB	CLD		487B-	0B	PHF	48E5-	1F	???	???						
4823-		3F	???		487C-	4D	49	69	EOR	##\$6949	48E6-	4B	PHA				
4824-		DF	???		487F-	09	05	ORA	##\$05	48E7-	0D	0D	3B	ORA	##\$3B0D		
4825-		DB	???		4881-	1B	CLC			48EA-	FF	???	???				
4826-		DF	???		4882-	DF	???	48EB-	3F	???	???						
4827-		03	???		4883-	FB	???	48EC-	0C	???	???						
4828-		4B	PHA		4884-	DB	???	48ED-	4D	29	DB	EOR	##\$DB29				
4829-	49	4D	EOR	##\$4D	4885-	1B	???	48F0-	1F	???	???						
482B-	69	6D	ADC	##\$6D	4886-	07	???	48F1-	1F	???	???						
482D-		DB	CLD		4887-	4B	PHA	48F2-	4B	PHA							
482E-		FF	???		4888-	2D	2D	6D	AND	##\$6D2D	48F3-	69	01	ADC	##\$01		
482F-		FB	???		488B-	09	0D	ORA	##\$0D	48F5-	00	BRK					
4830-		DF	???		488D-	DB	CLD			48F6-	DB	???	???				
4831-		DB	???		488E-	3B	???	48F7-	DB	???	???						
4832-		4B	PHA		488F-	DF	???	48F8-	93	???	???						
4833-	49	09	EOR	##\$09	4890-	DB	???	48F9-	92	???	???						
4835-	4D	49	01	EOR	##\$0149	4891-	DF	???	48FA-	92	???	???					
4838-		4B	PHA		4892-	4B	PHA	48FB-	2D	2D	2D	AND	##\$2D2D				
4839-		4B	PHA		4893-	09	2D	ORA	##\$2D	48FE-	2D	2D	2D	AND	##\$2D2D		
483A-		DB	???		4895-	2D	4D	09	AND	##\$094D	4901-	2D	3C	3F	AND	##\$3F3C	
483B-		92	???		4898-	00	BRK			4904-	3F	???	???				
483C-		92	???		4899-	DB	???	4905-	3F	???	???						
483D-		52	???		489A-	9B	???	4906-	3F	???	???						
483E-	09	2D	ORA	##\$2D	489B-	92	???	4907-	3F	???	???						
4840-	2D	6D	49	AND	##\$496D	489C-	92	???	4908-	3F	???	???					
4843-		DB	CLD		489D-	49	6D	EOR	##\$6D	4909-	27	???	???				
4844-		3B	???		489F-	2D	0D	18	AND	##\$180D	490A-	2D	2D	2D	AND	##\$2D2D	
4845-		DF	???		48A2-	DF	???	48A2-	DF	???	???	490D-	2D	2D	2D	AND	##\$2D2D
4846-		DB	???		48A3-	FB	???	48A3-	FB	???	???	4910-	2D	3C	3F	AND	##\$3F3C
4847-		FB	???		48A4-	FB	???	48A4-	FB	???	???	4913-	3F	???	???		
4848-		0B	PHF		48A5-	0B	PHF			4914-	3F	???	???				
4849-	4D	49	49	EOR	##\$4949	48A6-	4D	49	49	EOR	##\$4949	4915-	3F	???	???		
484C-	69	18	ADC	##\$18	48A9-	FC	???	48A9-	FC	???	???	4916-	3F	???	???		
484E-		DF	???		48AA-	DB	???	48AA-	DB	???	???	4917-	3F	???	???		
					48AB-	DB	???	48AB-	DB	???	???	4918-	27	???	???		
					48AC-	3B	???	48AC-	3B	???	???	4919-	2D	2D	6D	AND	##\$6D2D
					48AD-	6B	PLA			491C-	09	2D	ORA	##\$2D			
					48AE-	49	49	EOR	##\$49	491E-	2D	25	3F	AND	##\$3F25		

APPLE II

Seguito listato Tappeto mobile.

4921-	3F	???			4990-	DB	???			49FF-	49 49	EOR	##49
4922-	FF	???			4991-	6C 49 49	JMP	(\$4949)		4A01-	21 DF	AND	(\$DF, X)
4923-	1F	???			4994-	49 49	EOR	##49		4A03-	DB	???	
4924-	3F	???			4996-	49 49	EOR	##49		4A04-	DB	???	
4925-	3F	???			4998-	21 DF	AND	(\$DF, X)		4A05-	DB	???	
4926-	27	???			499A-	DB	???			4A06-	DB	???	
4927-	2D 2D 6D	AND	\$6D2D		499B-	DB	???			4A07-	DB	???	
492A-	0D 2D 2D	DRA	\$2D2D		499C-	DB	???			4A08-	DB	???	
492D-	25 3F	AND	\$3F		499D-	DB	???			4A09-	6C 49 49	JMP	(\$4949)
492F-	3F	???			499E-	DB	???			4A0C-	49 49	EOR	##49
4930-	FF	???			499F-	DB	???			4A0E-	49 49	EOR	##49
4931-	1F	???			49A0-	6C 49 49	JMP	(\$4949)		4A10-	21 DF	AND	(\$DF, X)
4932-	3F	???			49A3-	49 49	EOR	##49		4A12-	DB	???	
4933-	3F	???			49A5-	49 49	EOR	##49		4A13-	DB	???	
4934-	27	???			49A7-	21 DF	AND	(\$DF, X)		4A14-	DB	???	
4935-	2D 2D 6D	AND	\$6D2D		49A9-	DB	???			4A15-	DB	???	
4938-	0D 49 49	DRA	\$4949		49AA-	DB	???			4A16-	DB	???	
493B-	25 3F	AND	\$3F		49AB-	DB	???			4A17-	DB	???	
493D-	FF	???			49AC-	DB	???			4A18-	6C 49 49	JMP	(\$4949)
493E-	1B	???			49AD-	DB	???			4A1B-	49 49	EOR	##49
493F-	DF	???			49AE-	DB	???			4A1D-	49 49	EOR	##49
4940-	3B	???			49AF-	6C 49 49	JMP	(\$4949)		4A1F-	21 3F	AND	(\$3F, X)
4941-	3F	???			49B2-	49 49	EOR	##49		4A21-	3F	???	
4942-	2C 2D 6D	BIT	\$6D2D		49B4-	49 49	EOR	##49		4A22-	3F	???	
4945-	2D 0D 2D	AND	\$2D0D		49B6-	21 DF	AND	(\$DF, X)		4A23-	3F	???	
4948-	2D 3C 3F	AND	\$3F3C		49B8-	DB	???			4A24-	3F	???	
494B-	3F	???			49B9-	DB	???			4A25-	3F	???	
494C-	1F	???			49BA-	DB	???			4A26-	3F	???	
494D-	1F	???			49BB-	DB	???			4A27-	3F	???	
494E-	3F	???			49BC-	DB	???			4A28-	3F	???	
494F-	3F	???			49BD-	DB	???			4A29-	3F	???	
4950-	27	???			49BE-	6C 49 49	JMP	(\$4949)		4A2A-	27	???	
4951-	2D 2D 2D	AND	\$2D2D		49C1-	49 49	EOR	##49		4A2B-	00	BRK	
4954-	0D 2D 2D	DRA	\$2D2D		49C3-	49 49	EOR	##49		4A2C-	00	BRK	
4957-	2D 3C 3F	AND	\$3F3C		49C5-	21 DF	AND	(\$DF, X)					
495A-	3F	???			49C7-	DB	???						
495B-	3F	???			49C8-	DB	???						
495C-	3F	???			49C9-	DB	???						
495D-	3F	???			49CA-	DB	???						
495E-	3F	???			49CB-	DB	???						
495F-	27	???			49CC-	DB	???						
4960-	2D 2D 2D	AND	\$2D2D		49CD-	6C 49 49	JMP	(\$4949)					
4963-	2D 2D 2D	AND	\$2D2D		49D0-	49 49	EOR	##49					
4966-	2D 3C 3F	AND	\$3F3C		49D2-	49 49	EOR	##49					
4969-	3F	???			49D4-	21 DF	AND	(\$DF, X)					
496A-	3F	???			49D6-	DB	???						
496B-	3F	???			49D7-	DB	???						
496C-	3F	???			49D8-	DB	???						
496D-	3F	???			49D9-	DB	???						
496E-	27	???			49DA-	DB	???						
496F-	2D 2D 2D	AND	\$2D2D		49DB-	DB	???						
4972-	2D 2D 2D	AND	\$2D2D		49DC-	6C 49 49	JMP	(\$4949)					
4975-	2D 04 00	AND	\$0004		49DF-	49 49	EOR	##49					
4978-	DB	???			49E1-	49 49	EOR	##49					
4979-	DB	???			49E3-	21 DF	AND	(\$DF, X)					
497A-	DB	???			49E5-	DB	???						
497B-	93	???			49E6-	DB	???						
497C-	92	???			49E7-	DB	???						
497D-	92	???			49E8-	DB	???						
497E-	92	???			49E9-	DB	???						
497F-	2D 2D 2D	AND	\$2D2D		49EA-	DB	???						
4982-	2D 2D 2D	AND	\$2D2D		49EB-	6C 49 49	JMP	(\$4949)					
4985-	2D 2D 2D	AND	\$2D2D		49EE-	49 49	EOR	##49					
4988-	2D 25 DF	AND	\$DF25		49F0-	49 49	EOR	##49					
498B-	DB	???			49F2-	21 DF	AND	(\$DF, X)					
498C-	DB	???			49F4-	DB	???						
498D-	DB	???			49F5-	DB	???						
498E-	DB	???			49F6-	DB	???						
498F-	DB	???			49F7-	DB	???						
					49F8-	DB	???						
					49F9-	DB	???						
					49FA-	6C 49 49	JMP	(\$4949)					
					49FD-	49 49	EOR	##49					





**GRUPPO EDITORIALE
JACKSON**

DIVISIONE LIBRI



Scopri il tuo home computer

A. Borra, M.C. Grizzi
**LAVORIAMO CON
IL COMMODORE 16**

Lavorare con il C16 è un piacere, fallo anche tu, ti divertirai.

Cod. CC244 Pag. 196 L. 20.000
Con cassetta

G. Marano
**30 PROGRAMMI
PER MSX**

Inserite soltanto la cassetta e... una valanga di programmi uscirà dal vostro MSX.

Cod. CZ181 Pag. 96 L. 20.000
Con cassetta

A. Cattaneo, P. Todorovich
COMPUTER HARDWARE

Il primo vero libro di realizzazioni pratiche dedicate ai più diffusi home computer: C64, Spectrum, VIC20, C16 e MSX.

Cod. CH182 Pag. 152 L. 13.000

A.P. Stephenson
**IL COMPUTER:
l'Hobby e il lavoro**

Il computer come compagno ideale, nella propria professione e nell'hobby.

Cod. CI187 Pag. 192 L. 12.000

*R. Bonelli, L. Pazzucconi,
F. Racchi*
**GUIDA AL COMMODORE
PLUS 4**

Un manuale completo per il Plus 4 ripercorrendo C16 per te e C16 sempre di più.

Cod. CC256 Pag. 470 L. 30.000



**GRUPPO EDITORIALE
JACKSON**
DIVISIONE LIBRI

ritagliare (o fotocopiare) e spedire in busta chiusa a
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
Totale			

Pagherò contrassegno al postino il prezzo indicato più L. 3.000 per contributo fisso spese di spedizione.

Condizioni di pagamento con esenzione del contributo spese di spedizione:

Allego assegno della Banca Allego fotocopia del versamento su c/c n. 11666203 a voi intestato

Allego fotocopia di versamento su vaglia postale a voi intestato

n° _____

Nome _____

Cognome _____

Via _____

Cap. _____ Città _____ Prov. _____

Data _____ Firma _____

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A. _____

ORDINE
MINIMO
L. 50.000

**Jackson
la biblioteca
che fa
testo**

Music utility

Completiamo Music editor con l'aggiunta di un'interessante routine

di Tommaso Palermo

La prima critica ricevuta per il programma "Music editor" (M.e. d'ora in poi, vedere **Personal Software** n. 33, Novembre '85) riguardava il fatto che i programmi musicali da questo generati sono finiti a se stessi e non possono essere utilizzati per sonorizzare altri programmi. Confessiamo che proprio questa era la nostra intenzione: suggestionati dall'ascolto sul Commodore 64 di alcuni programmi in grado di eseguire splendide trascrizioni di Bach, avevamo faticosamente costruito, mediante un editor piuttosto rudimentale, un programma che eseguiva un intero concerto di Vivaldi. Da questa esperienza era nata l'idea di M.e., che non aveva però tra i suoi scopi quello di sonorizzare programmi altrui, bensì quello di formare una raccolta musicale per computer, o, perché no, di dare all'utente la possibilità di cimentarsi nella composizione anche senza avere completa padronanza di uno strumento.

Sollecitati comunque dall'appunto di cui sopra, abbiamo ideato questa routine (listato 1) che permetterà d'inserire le musiche trascritte per mezzo di M.e. nei propri programmi, senza rallentare l'esecuzione mentre il brano viene suonato e, soprattutto - nel limite, piuttosto ampio, di 1.023 note per brano - senza sottrarre memoria Ram al Basic.

Chi già conosce il funzionamento del C 64, avrà senz'altro compreso come ciò può avvenire: per quanto riguarda il primo punto, affidando l'esecuzione anziché a una routine in Basic - come in M.e. - a una routine in linguaggio macchina richiamata dall'interrupt; riguardo al secondo punto invece, i dati delle note vengono trascritti nella Ram nascosta sotto gli 8 Kbyte dell'interprete Basic (Ram che non può naturalmente essere adoperata dal Basic), da \$A000 a \$BFFF. Poiché la codifica di ogni nota - per ognuna delle tre voci - richiede 8 byte, ne risulta che questa poco usata area di memoria è sufficiente a contenere 1.024 note meno una (l'ultima segnala la fine del brano).

A cosa serve l'utility

Applicata, nel modo che vedremo, a uno dei programmini musicali generati da M.e., Music utility creerà un ulteriore nuovo programma Basic. Quest'ultimo,

il programma richiamante, avrà come prima riga la seguente:

```
10 IF MU=0 THEN MU=49173:LOAD
   "[NOME PROG.MUSICALE]",8,1
```

Dopodiché, in qualsiasi punto del programma si voglia fare iniziare l'esecuzione, basterà scrivere l'istruzione Sysmu (o Sys 49173); altri comandi, applicabili all'esecuzione musicale, sono inoltre:

- Sys 49391 (interrompe l'esecuzione);
- Sys 49202 (riprende l'esecuzione dal punto in cui era stata eventualmente interrotta);
- Wait 788,1 (sospende l'esecuzione del programma fino a che non cessa la musica).

Come si può vedere non si tratta di un

Modalità d'uso

Per creare questo nuovo programma bisogna:

- 1) caricare in memoria il programma musicale (quello ottenuto, in M.e., dal comando Ctrl X, dopo ovviamente che il brano è stato trascritto);
- 2) modificare, come in figura 2, la linea 1 del programma, facendo attenzione a non alterarne la lunghezza! (Per provare il funzionamento basta dare Run2: se la musica viene suonata, bene, altrimenti si riprova, tenendo conto che Load e Rem occupano entrambe un solo byte nella riga Basic);
- 3) inserire nel drive un dischetto contenente il programma Music utility e, possibilmente, lo spazio per il nuovo programma;

1	REM *****	1	LOAD "MUSIC UTILITY",8:*****
2	REM *	2	REM *
3	REM * J.S. BACH	3	REM * J.S. BACH
4	REM * PRELUDIO N.1	4	REM * PRELUDIO N.1
5	REM *	5	REM *
6	REM *****	6	REM *****

Figura 1 - La differenza tra le testate di Music editor e Music utility.

set completo d'istruzioni per il trattamento della colonna sonora, ma è quanto basta per corredare i propri programmi di un gradevole sottofondo musicale con effetti distensivi... Chi lavora con l'Easy Script sa, per esempio, che premendo F1 e Ctrl 3 può continuare a scrivere per un minutino circa accompagnato dall'esecuzione di un brano di Elgar; a un'esigenza del genere risponde perfettamente la nostra utility.

4) battere Run.

Queste operazioni sono indispensabili per mantenere protetta l'area su cui sono scritti i dati delle note, nonché per permettere a Music utility di riconoscere da sé fino a dove queste sono trascritte. Gli altri dati invece verranno richiesti dal programma, dopo che questo è stato caricato e dopo aver trascritto i dati in linguaggio macchina da 49173 a 49437.

Bisogna quindi inserire, nell'ordine:

Listato 1 - Il programma principale Music utility.

```

10 REM***UTILITY PER IL MUSIC EDITOR (PS
    NOV.85)***DI TOMMASO PALERMO***
20 FORI=49173TO49437:READA:POKEI,A:T=T+A
:NEXT
30 IFT<>33767THENPRINT"{RVS ON}ERRORE NE
I DATA!":END
40 INPUT "{CLR}{CUR.GIU}TITOLO DEL BRANO
";BR$
50 FORI=0TO2:A=49152+7*I
60 PRINTTAB(5)"{CUR.GIU}{RVS ON}{2 SPC}V
OCE "STR$(I+1)" {CUR.GIU}"
70 INPUT"ATTACK-DECAY{4 SPC}";D:IFD>255T
HEN70
80 POKEA+5,D
90 INPUT"SUSTAIN-RELEASE ";S:IFS>255THEN
90
100 POKEA+6,S
110 INPUT"WAVEFORM ({RVS ON}T{RVS OFF},{
RVS ON}S{RVS OFF},{RVS ON}P{RVS OFF}
,{RVS ON}N{RVS OFF})";B$
120 IFB$="T"THENB=17:GOTO170
130 IFB$="S"THENB=33:GOTO170
140 IFB$="P"THENB=65:GOTO170
150 IFB$="N"THENB=129:GOTO170
160 GOTO110
170 POKEA+4,B:IFB<>65THEN200
180 INPUT"DUTY CYCLE (0-4095)";B:IFB>409
5THEN180
190 POKEA+3,B/256:POKEA+2,(BAND255)
200 POKEA,0:POKEA+1,0:NEXT
210 INPUT "{CUR.GIU}{RVS ON} VELOCITA' {
RVS OFF} (F:DA 1.0 A 3.5)";V
220 IFV<1ORV>3.5THEN210
230 PRINT"{CUR.GIU}"TAB(12)"{RVS ON} ATT
ENDI " :A=49152:X=PEEK(46)*256+PEEK(4
5)-8
240 FORI=4096TOX
250 A=A-1:C=PEEK(I):IFJ=0THENC=C*V+.4:IF
C>255THENC=255
260 J=(J+1)AND7:POKEA,C:NEXT
270 POKEA,0:PRINTA

```

```

275 POKE198,0:PRINT"{CUR.GIU}{RVS ON}{2
SPC}PREMI UN TASTO QUANDO SEI PRONTO
{2 SPC}":WAIT198,1
280 POKE252,A/256:POKE251,A-PEEK(252)*25
6:POKE255,193:POKE254,203
290 OPEN1,8,1,BR$:POKE253,10:SYS49418:CL
OSE1
292 IF(PEEK(253)<10ANDPEEK(253)>0)THENPR
INT"ERRORE DI REGISTRAZIONE!":END
295 PRINT"{RVS ON}SYS49173{RVS OFF} = SU
ONA * {RVS ON}SYS49391{RVS OFF} = IN
TERROMPE":SYS49173
300 DATA162,0,189,0,192,157,0,212,232,22
4,21,208,245,169,192,133,252,169,0
310 DATA133,251,169,0,133,253,169,1,133,
254,169,15,141,24,212,120,169,72,141
320 DATA20,3,169,192,141,21,3,88,96,234,
234,234,234,72,138,72,152,72,198
330 DATA254,208,114,165,1,41,254,133,1,1
64,253,208,2,198,252,136,177,251,240
340 DATA105,133,254,136,177,251,141,1,21
2,136,177,251,141,0,212,136,177,251,
141
350 DATA8,212,136,177,251,141,7,212,136,
177,251,141,15,212,136,177,251,141,1
4
360 DATA212,136,177,251,170,41,1,240,5,1
69,0,141,4,212,138,41,2,240,5
370 DATA169,0,141,11,212,138,41,4,240,5,
169,0,141,18,212,132,253,165,1
380 DATA9,1,133,1,173,4,192,141,4,212,17
3,11,192,141,11,212,173,18,192
390 DATA141,18,212,104,168,104,170,104,7
6,49,234,141,4,212,141,11,212,141,18
400 DATA212,141,24,212,169,49,141,20,3,1
69,234,141,21,3,165,1,9,1,133
410 DATA1,104,168,104,170,104,76,49,234,
169,0,141,4,212,141,11,212,141,18
420 DATA212,141,24,212,120,169,49,141,20
,3,169,234,141,21,3,88,96
500 DATA169,54,133,1,169,251,166,254,164
,255,32,216,255,133,253,169,55,133,1
,96

```

a) il nome sotto cui va registrato il nuovo programma;

b) i dati dell'involuppo per ognuna delle tre voci (occorre ricordare quelli impostati in M.e., che comunque sono facilmente ricavabili dal listato del programma precedente), che saranno trascritti da 49152 a 49172;

c) la velocità d'esecuzione.

Sul punto c) occorre fare una nota: la velocità, nel programma originale, è impostata dalla variabile F, che si legge alla fine della riga 10: a minor valore di F corrisponde maggiore velocità d'esecuzione. Però il ritmo calcolato teneva conto anche dei rallentamenti operati dal Basic, che non esistono invece nel

nuovo programma; in linea di massima si può dire che per valori di F medi o alti (da due in su), questa accelerazione è meno sensibile e non pone grossi problemi; per valori più bassi, oltre a notarsi la differenza di velocità, potrebbero verificarsi anche degli scompensi nel ritmo; sarà perciò opportuno aumentare il valore che si introduce rispetto a quello che compare nel listato del programma musicale in Basic.

Terminata l'introduzione di questi dati e al termine dell'elaborazione da parte del computer, appare scritto un numero (l'indirizzo di partenza del nuovo programma: verificare che non sia più basso di 40.960, che non sia cioè occupata

la Ram del Basic) e l'invito a premere un tasto; dopo averlo premuto, il nuovo programma viene registrato col nome che gli era stato scelto all'inizio, ed è praticamente pronto all'uso di cui sopra.

Come funziona

Su questo punto, rimandiamo soprattutto alle note che accompagnano il disassemblato della routine in linguaggio macchina (listato 2), dal momento che il programma Basic svolge un compito quasi esclusivamente da "piccolo scrivano": trascrive la routine in lin-

COMMODORE 64

guaggio macchina da \$C015 a \$C011 (righe 20-30), i valori da inserire nel Sid da \$C000 a \$C014 (50-200), e i codici delle note da \$BFFF in giù (210-270); e infine (col suo unico atto "creativo") fa funzionare la routine di registrazione (275-295).

Il programma non può essere ampliato, dal momento che così come è riportato nel listato 1, occupa la memoria da

2049 a 4078, e i dati delle note musicali nel programma originale sono trascritti da 4086 in su, con un margine di elasticità di soli 18 byte. Chi non dispone di drive, oltre a cambiare in uno tutti gli otto degli esempi qui riportati, deve anche modificare la linea 290 (e ricordarsi di registrare il programma musicale a seguire, nella cassetta, il programma richiamante): la ristrettezza di memoria

non ha permesso d'inserire opzioni in Music utility.

Ricordiamo infine che, trovandosi la routine d'esecuzione scritta tra 49152 e 49417, non è compatibile con altre routine in linguaggio macchina che dovessero risiedere nella stessa area di memoria; è compatibile invece con le routine del Turbo tape, che sono trascritte a partire dall'indirizzo 50000. ■

```
{HOME}.....HOME
{CLR}.....PULIZIA SCHERMO
{CUR.SU}.....CURSORE IN ALTO
{CUR.GIU}.....CURSORE IN BASSO
{CUR.DES}.....CURSORE A DESTRA
{CUR.SIN}.....CURSORE A SINISTRA
{SPC}.....SPAZIO
{RVS ON}.....REVERSE ON
{RVS OFF}.....REVERSE OFF
{INST}.....INSERT
{F1}.....TASTO F1
{F2}.....TASTO F2
{F3}.....TASTO F3
{F4}.....TASTO F4
{F5}.....TASTO F5
{F6}.....TASTO F6
{F7}.....TASTO F7
{F8}.....TASTO F8
{BLACK}.....COL. NERO (CTRL+1)
{WHITE}.....COL. BIANCO (CTRL+2)
{RED}.....COL. ROSSO (CTRL+3)
{CYAN}.....COL. CIANO (CTRL+4)
```

```
{PURPLE}.....COL. PORPORA (CTRL+5)
{GREEN}.....COL. VERDE (CTRL+6)
{BLUE}.....COL. BLU (CTRL+7)
{YELLOW}.....COL. GIALLO (CTRL+8)
{ORANGE}.....COL. ARANCIO (CBM+1)
{BROWN}.....COL. MARRONE (CBM+2)
{LT.RED}.....COL. ROSSO CHIARO (CBM+3)
{GRAY1}.....COL. GRIGIO 1 (CBM+4)
{GRAY2}.....COL. GRIGIO 2 (CBM+5)
{LT.GREEN}.....COL. VERDE CHIARO (CBM+6)
{LT.BLUE}.....COL. BLU CHIARO (CBM+7)
{GRAY3}.....COL. GRIGIO 3 (CBM+8)
```

I CARATTERI GRAFICI, OTTENUTI CON LA PRESSIONE DEI TASTI 'SHIFT' E 'CBM', SONO CODIFICATI IN MODO DA INDICARE IL TASTO DA PREMERE ASSIEME A 'SHIFT' O 'CBM'. ES. IL CUORICINO E' CODIFICATO CON {SH S}.

UN NUMERO DENTRO LE PARENTESI INDICA LE VOLTE CHE IL TASTO VA PREMUTO.

Figura 2 - Tabella di conversione.

Listato 2 - La routine in linguaggio macchina.

```
c015 a2 00      ldx ##00      ;-----
c017 bd 00 c0   lda $c000,x  ;Inizializza
c01a 9d 00 d4   sta $d400,x  ;i registri
c01d e8         inx          ;del SID,co-
c01e e0 15      cpx ##15     ;piandoli da
c020 d0 f5      bne $c017    ;c000-c014.
c022 a9 c0      lda #$c0     ;-----
c024 85 fc      sta $fc      ;Setta i va-
c026 a9 00      lda ##00     ;lori delle
c028 85 fb      sta $fb      ;locazioni
c02a a9 00      lda ##00     ;in pagina 0
c02c 85 fd      sta $fd      ;usate come
c02e a9 01      lda #$01     ;contatori.
c030 85 fe      sta $fe      ;-----
c032 a9 0f      lda #$0f     ;Accende il
c034 8d 18 d4   sta $d418   ;volume.
c037 78         sei          ;-----
c038 a9 48      lda #$48     ;Modifica il
c03a 8d 14 03   sta $0314   ;vettore
c03d a9 c0      lda #$c0     ;d'IRQ
c03f 8d 15 03   sta $0315   ;
```

```
c042 58         cli          ;
c043 60         rts          ;-----
c044 ea         nop          ;-----
c045 ea         nop          ;Nuova
c046 ea         nop          ;interrupt
c047 ea         nop          ;routine:
c048 48         pha          ;-----
c049 8a         txa          ;Salva i
c04a 48         pha          ;registri
c04b 98         tya          ;nello Stack
c04c 48         pha          ;-----
c04d c6 fe      dec $fe      ;Decrementa
c04f d0 72      bne $c0c3   ;il tempo.
c051 a5 01      lda $01     ;-----
c053 29 fe      and #$fe     ;Attiva RAM
c055 85 01      sta $01     ;a000-bfff
c057 a4 fd      ldy $fd     ;-----
c059 d0 02      bne $c05d   ;Aggiorna i
c05b c6 fc      dec $fc     ;contatori.
c05d 88         dey          ;-----
c05e b1 fb      lda ($fb),y ;Preleva il
c060 f0 69      beq $c0cb   ;tempo (0=
c062 85 fe      sta $fe     ;FINE).
```

Seguito listato 2.

```

c064 88      dey      ;-----
c065 b1 fb   lda ($fb),y ;VOCE 1:
c067 8d 01 d4 sta $d401 ;High-Byte
c06a 88      dey      ;
c06b b1 fb   lda ($fb),y ;
c06d 8d 00 d4 sta $d400 ;Low-Byte
c070 88      dey      ;-----
c071 b1 fb   lda ($fb),y ;VOCE 2:
c073 8d 08 d4 sta $d408 ;High-Byte
c076 88      dey      ;
c077 b1 fb   lda ($fb),y ;
c079 8d 07 d4 sta $d407 ;Low-Byte
c07c 88      dey      ;-----
c07d b1 fb   lda ($fb),y ;VOCE 3:
c07f 8d 0f d4 sta $d40f ;High-Byte
c082 88      dey      ;
c083 b1 fb   lda ($fb),y ;
c085 8d 0e d4 sta $d40e ;Low-Byte
c088 88      dey      ;-----
c089 b1 fb   lda ($fb),y ;Preleva il
c08b aa      tax      ;il byte di
c08c 29 01   and #$010 ;controllo
c08e f0 05   beq $c095 ;in base a
c090 a9 00   lda #$00 ;cui vengono
c092 8d 04 d4 sta $d404 ;settati i
c095 8a      txa      ;registri di
c096 29 02   and #$02 ;waveform.
c098 f0 05   beq $c09f ;
c09a a9 00   lda #$00 ;
c09c 8d 0b d4 sta $d40b ;
c09f 8a      txa      ;
c0a0 29 04   and #$04 ;
c0a2 f0 05   beq $c0a9 ;
c0a4 a9 00   lda #$00 ;
c0a6 8d 12 d4 sta $d412 ;-----
c0a9 84 fd   sty $fd ;Salva il
c0ab a5 01   lda $01 ;contatore e
c0ad 09 01   ora #$01 ;attiva ROM
c0af 85 01   sta $01 ;a000-bfff
c0b1 ad 04 c0 lda $c004 ;-----
c0b4 8d 04 d4 sta $d404 ;'Accende'
c0b7 ad 0b c0 lda $c00b ;ognuna del-
c0ba 8d 0b d4 sta $d40b ;le tre voci
c0bd ad 12 c0 lda $c012 ;
c0c0 8d 12 d4 sta $d412 ;-----
c0c3 68      pla      ;Preleva i
c0c4 a8      tay      ;registri
c0c5 68      pla      ;dallo Stack
c0c6 aa      tax      ;e salta ad
c0c7 68      pla      ;IRGK.
c0c8 4c 31 ea jmp $ea31 ;-----
c0cb 8d 04 d4 sta $d404 ;FINE brano
c0ce 8d 0b d4 sta $d40b ;'spegne' il
c0d1 8d 12 d4 sta $d412 ;SID, ripri-
c0d4 8d 18 d4 sta $d418 ;stina il
c0d7 a9 31   lda #$31 ;vettore di
c0d9 8d 14 03 sta $0314 ;IRG,riatti-
c0dc a9 ea   lda #$ea ;va la ROM,
c0de 8d 15 03 sta $0315 ;e salta in-
c0e1 a5 01   lda $01 ;fine ad

```

```

c0e3 09 01   ora #$01 ;IRGK.
c0e5 85 01   sta $01 ;
c0e7 68      pla      ;
c0e8 a8      tay      ;
c0e9 68      pla      ;
c0ea aa      tax      ;
c0eb 68      pla      ;
c0ec 4c 31 ea jmp $ea31 ;-----
c0ef a9 00   lda #$00 ;Duplica in
c0f1 8d 04 d4 sta $d404 ;parte la
c0f4 8d 0b d4 sta $d40b ;routine
c0f7 8d 12 d4 sta $d412 ;precedente.
c0fa 8d 18 d4 sta $d418 ;Serve a in-
c0fd 78      sei      ;terrompere
c0fe a9 31   lda #$31 ;il brano
c100 8d 14 03 sta $0314 ;'dall'e-
c103 a9 ea   lda #$ea ;sterno'
c105 8d 15 03 sta $0315 ;(Sys 49391)
c108 58      cli      ;
c109 60      rts      ;
c10a a9 36   lda #$36 ;-----
c10c 85 01   sta $01 ;Routine di
c10e a9 fb   lda #$fb ;registra-
c110 a6 fe   ldx $fe ;zione: non
c112 a4 ff   ldy $ff ;fara' parte
c114 20 d8 ff jsr $ffd8 ;del pro-
c117 85 fd   sta $fd ;gramma de-
c119 a9 37   lda #$37 ;finitivo.
c11b 85 01   sta $01 ;
c11d 60      rts      ;-----

```

Listato 3 - Un esempio di esecuzione sonora.

```

1 REM*****
2 REM      ANTONIO VIVALDI
3 REM      CONCERTO PER FLAUTINO IN DO MAG.
4 REM      OP.44,11
5 REM
6 REM
7 REM      ARRANGIAMENTO E PROGRAMMAZIONE
8 REM      DI TOMMASO PALERMO
9 REM*****
15 POKE53280,6:POKE53281,6:PRINT"{CLR}{Y
   ELLOW}{6 CUR.GIU}"
20 PRINTTAB(14)"V I V A L D I{4 CUR.GIU}
    "
25 PRINTTAB(16)"OP. 44,11{4 CUR.GIU}"
30 PRINTTAB(10)"CONCERTO PER FLAUTINO{CU
    R.GIU}"
35 PRINTTAB(8)"{BLUE}PRIMO MOVIMENTO : A
    LLEGRO{YELLOW}":GOSUB190
40 L1=54272:H1=L1+1:W1=L1+4:A1=L1+5:S1=L
    1+6
42 L2=54279:H2=L2+1:W2=L2+4:A2=L2+5:S2=L
    2+6
44 L3=54286:H3=L3+1:W3=L3+4:A3=L3+5:S3=L
    3+6
46 POKE54296,15:A=3841:B=16241:F=4
50 GOSUB270:GOSUB100:GOSUB200
60 PRINT"{BLUE}{CUR.SU}";PRINTTAB(8)"SE
    CONDO MOVIMENTO : LARGO{YELLOW}":GOSU
    B190

```

COMMODORE 64

Seguito listato 3.

```

65 F=1:POKEA1,127:POKEA2,106:POKEA3,106:
   POKES1,79:POKES2,64:POKES3,64
66 A=16250:B=17002:GOSUB100:A=16250:B=17
   002:GOSUB100
67 A=17018:B=18458:GOSUB100:A=17018:B=18
   466:GOSUB100:GOSUB200
70 PRINT"{CUR.SU}{BLUE}";:PRINTTAB(9)"FI
   NALE : ALLEGRO{2 SPC}ASSAI{YELLOW}":F
   =4:GOSUB190
75 A=18640:B=29792:GOSUB100
99 POKE54296,0:END
100 FORI=ATOBSTEP8
110 TI$="000000"
120 IF(PEEK(I+7)AND1)THENPOKEW1,0:POKEH1
   ,PEEK(I+1):POKEL1,PEEK(I+2):POKEW1,W
   W
130 IF(PEEK(I+7)AND2)THENPOKEW2,0:POKEH2
   ,PEEK(I+3):POKEL2,PEEK(I+4):POKEW2,3
   3
140 IF(PEEK(I+7)AND4)THENPOKEW3,0:POKEH3

```

```

   ,PEEK(I+5):POKEL3,PEEK(I+6):POKEW3,3
   3
145 IF(PEEK(I+7)AND8)THENGOSUB250
150 IFTI<PEEK(I)*FTHEN150
160 NEXT:RETURN
190 FORI=56050TO56150:POKEI,7:FORT=0TO30
   :NEXT:NEXT:RETURN
200 FORI=1780TO1900:POKEI,32:FORT=0TO30:
   NEXT:NEXT:RETURN
250 R=R+1
255 ONRGOTO260,270,280,290,270,260,270,2
   60,280,290,270,260,270,260,300,305,2
   70
260 POKEA1,127:POKES1,79:WW=17:RETURN
270 POKEA1,106:POKEA2,106:POKEA3,26:POKE
   S1,32 :POKES2,32 :POKES3,170:WW=33:R
   ETURN
280 POKEA1,127:POKES1,79:WW=17:POKEA3,10
   6:POKES3,32:RETURN
290 POKEA3,26:POKES3,170:RETURN
300 F=6:RETURN
305 F=4:R=12:RETURN

```

Totocalcio per Msx (Personal Software n. 35)

L'istruzione Width nel Basic Msx

Come molti di voi forse già sapranno, il Basic Msx permette di dimensionare la grandezza del video scegliendo il numero di colonne con cui deve essere stampato il testo nei modi Screen 0 e Screen 1.

Questa istruzione è Width (n), dove n è appunto il numero di colonne che deve essere scelto tra un massimo di 40 per il modo 0 e un massimo di 32 per il modo 1.

Purtroppo però all'accensione del computer, non tutte le marche si comportano allo stesso modo: per esempio lo Spectravideo SVI 728 si pone in modo Screen 0 con 39 colonne di testo, mentre i computer Philips VG 8000 partono con 37 colonne di testo.

Questo può comportare degli errori di incolonnamento quando si copia un listato o si carica un programma scritto con un computer Msx di marca diversa. È naturale che per ovviare a questo inconveniente basta aggiungere al listato, ove non sia già presente, l'istruzione Width con il giusto dimensionamento.

Paolo Ferrami

Grand Prix per C 64 (Personal Software n. 31)

Nel programma Grand Prix per C 64 la linea 500 va così modificata:

```

SI = 0:GOSUB 380:IF NOS$ = "N" THEN
OPEN 1,1,2,NFS:GOTO 550

```

PUOI ESSERE COSÌ CONTENTO CON IL TUO SOFTWARE?



Allora spedisci subito il tagliando per una prova **GRATIS** se vuoi facilmente imparare e creare tu stesso il software con il corso BASIC dell'IST.

IST La scuola del progresso

Via S. Pietro 49 - 21016 LUINO (VA) - Tel. 0332/53.04.69

Si, desidero ricevere - in visione gratuita, con invio raccomandato e senza alcun impegno - la prima dispensa in Prova di Studio per un periodo massimo di 3 settimane. Se io non Vi ritornerò la dispensa entro tale termine, sarò automaticamente iscritto al Vostro corso di BASIC e riceverò regolare conferma.

Cognome _____

Nome _____ Etá _____

Via _____ N. _____ CAP _____ Città _____

TAGLIANDO
79A

I SEGRETI DEI PERSONAL

di Sergio Borsani

Input function

Un giorno, tra amici, si discuteva sulla possibilità di ridefinire una funzione matematica con una formula immessa in input da un programma Basic. Come si sa, in questi casi, generalmente si deve interrompere l'esecuzione del programma, editare la linea contenente l'istruzione Def e dare nuovamente il Run. La procedura non è elegante, ma, in mancanza d'altro, diventa accettabile. Ovviamente non funziona quello che potrebbe sembrare l'uovo di Colombo: accettare la formula in input con la stringa A\$ e poi scrivere Def Y = A\$.

Il programma che segue riesce nell'intento modificando i byte che codificano la linea di programma contenente l'istruzione Def mediante delle Call Peek e Call Load. Per il suo funzionamento è richiesto l'Extended Basic e l'espansione di memoria; quest'ultima non perché il programma sia troppo voluminoso, ma perché con tale configurazione il programma invece di risiedere nella Vdp Ram viene trasferito nella high memory dove può essere facilmente manipolato.

Alla linea 100 viene definita la funzione $Y = X \star 2$; la linea 110 legge il valore dei puntatori alla Tavola dei numeri di linea, mentre le seguenti calcolano l'indirizzo di memoria della linea 100. Il ciclo alla linea 140 legge e stampa in forma decimale i byte che costituiscono la linea 100.

Se viene dato il Run, sullo schermo appaiono i nove numeri riportati al termine del listato. 137 è il token della parola riservata Def, 89 è il codice Ascii della lettera Y, 190 è il token del segno =, 88 corrisponde alla X, 195 al segno di moltiplicazione, seguono i codici 200 e 1 e infine 50 che è il codice Ascii del numero due. L'ultimo byte della linea è sempre impostato a zero, mentre all'indirizzo immediatamente precedente la linea stessa c'è un byte che ne indica la lunghezza. Il codice 200 precede sempre una costante, il valore uno che lo segue sta a indicare che tale costante è formata da una sola cifra. A esempio il numero dieci verrebbe codificato con la

sequenza 200, 2, 49, 48. Anche il punto decimale viene considerato alla stregua delle cifre.

È istruttivo cambiare la funzione presente nella linea 100, dare il Run e ottenere l'immagine della linea stessa così come si trova in memoria. Nel listato 2 la funzione è stata cambiata in $Y = \text{Sin}(X)$.

Noti i valori token delle varie funzioni matematiche: Int, Sin, Log, eccetera, si può usare l'istruzione Call Load per intervenire direttamente nella Ram. Ma attenzione, se la nuova funzione occupa più byte della precedente c'è il rischio di sovrapporre due linee del programma. Questo non succede durante l'editing usuale poiché in tal caso il sistema operativo si incarica di cambiare gli indirizzi e di spostare tutta la tavola dei numeri di linea. La figura 1 illustra cosa avviene quando si edita la linea 100 scrivendo una nuova funzione che occupa un byte di memoria in meno della precedente. Come si vede è cambiato l'indirizzo di linea ed è stata spostata di una locazione tutta la Tavola dei numeri di linea.

Il programma Basic in grado di emulare sotto questo aspetto il sistema operativo risulterebbe troppo lento, ma c'è il modo di aggirare l'ostacolo. La prima volta si

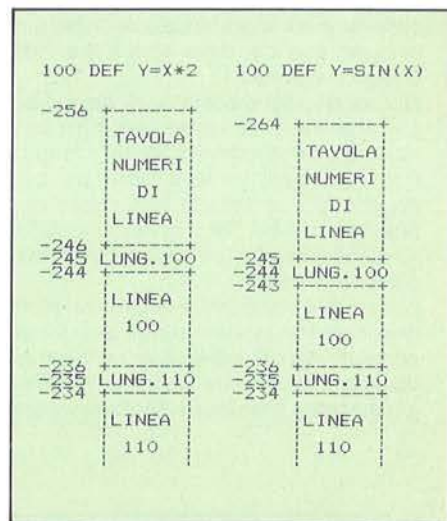


Figura 1 - Il sistema operativo gestisce in modo dinamico gli spazi riservati al programma e alla tavola dei numeri di linea. Editando una linea di programma, non solo cambia il suo indirizzo di partenza, ma viene allocata l'intera Tavola dei numeri di linea.

scrive una linea Def con una funzione che occupi quanto più spazio possibile, in que-

TEXAS TI99/4A

sto modo: Def $Y = X + X + X + X \dots$, fino a occupare cinque righe di schermo. Successivamente si accetta in input una stringa contenente una funzione, la si codifica con i valori token caricandola in memoria sopra la precedente e si modifica il byte che contiene la lunghezza della linea interessata. I caratteri eccedenti, cioè $X + X + X + \dots$, rimangono per così dire nascosti, non appaiono con un List e vengono del tutto ignorati quando viene calcolato il valore della funzione; essi vengono tuttavia conservati con i comandi Save e Old e restano a disposizione per scrivere formule più o meno lunghe.

Il programma del listato 3 è solo dimostrativo, ma può ben adattarsi ad altri pro-

grammi ove sia richiesta una tecnica analoga, a esempio quando si voglia disegnare il grafico di diverse funzioni senza dover interrompere l'esecuzione per ridefinirle. Segue una breve descrizione delle principali routine.

120 - Definizione della funzione.

130 - Dimensionamento delle matrici e caricamento dei valori token.

140-160 - Ricerca dell'indirizzo della linea 120.

170 - Input della funzione.

180-240 - Codifica della funzione con i valori token.

250-260 - Controllo della corrispondenza tra parentesi aperte e chiuse.

270 - Scrittura in memoria dei byte che

costituiscono la linea Def.

280-300 - Calcolo della funzione per alcuni valori di X.

310-330 - Data contenenti i valori token.

Il programma accetta solo funzioni nella variabile X ed è protetto contro alcuni errori, ma non possiede un controllo completo della sintassi. Sono errori recuperabili parole riservate diverse da Atn, Cos, Int, Log, Sin, Tan, Pi; errori ortografici, variabili diverse dalla X, parentesi spaiate; sono invece irrecuperabili errori nella struttura della formula. Se si scrive $F(X) = X.5 + \star/2$ il programma si arresta, compare un messaggio d'errore da parte del sistema e per riprendere bisogna dare nuovamente il Run. ■

Listato 1 - Un piccolo programma d'esempio e il suo output.

```
100 DEF Y=X*2
110 CALL PEEK(-31952,A,B,C,D):: A=A*256+
B-65536 :: C=C*256+D-65536
120 FOR I=C-3 TO A STEP -1 :: CALL PEEK(
I,B,D,E,F):: B=B*256+D :: IF B=100 THEN
140
130 NEXT I :: STOP
140 E=E*256+F-65536 :: I=E-1 :: CALL PEE
K(I,B):: FOR I=E TO E+B-1 :: CALL PEEK(I
,BY):: PRINT BY;:: NEXT I
RUN
137 89 190 88 195 200 1 50 0
```

Listato 2 - Sostituendo questa riga nel listato precedente si ha un nuovo output.

```
100 DEF Y=SIN(X)
.....
RUN
137 89 190 210 183 88 182 0
```

Listato 3 - Il programma dimostrativo.

```
120 DEF Y=X+X+X+X+X+X+X+X+X+X+X+X+X+X+
X+X+X+X+X+X+X+X+X+X+X+X+X+X+X+X+X+
X+X+X+X+X+X+X+X+X+X+X+X+X+X+X+X+X+
X+X+X+X+X+X+X+X+X+X+X+X+X+X+X+X+X+
130 CALL INIT :: DIM TV(256),TT$(15,2)::
FOR J=0 TO 15 :: READ TT$(J,1),TT$(J,2)
:: NEXT J
140 CALL PEEK(-31952,A,B,C,D):: Z=65536
:: A=A*256+B-Z :: C=C*256+D-Z :: I=C-3
150 FOR I=C-3 TO A STEP -4 :: CALL PEEK(
I,B,D,E,F):: B=B*256+D :: IF B=120 THEN
E=E*256+F-Z :: GOTO 170
```

```
160 NEXT I :: STOP
170 CALL CLEAR :: INPUT "F(X)=":FN$ :: P
RINT RPT$("-",28):: PRINT :: CTR=0 :: FG
=0
180 FOR J=1 TO LEN(FN$):: W$=SEG$(FN$,J,
1):: Q=ASC(W$):: IF (Q>47 AND Q<58)OR Q=
46 THEN 200 ELSE IF FG=1 THEN TV(CW)=NC
:: FG=0
190 FOR K=0 TO 15 :: P=POS(FN$,TT$(K,1),
J):: IF P<>J THEN 220 ELSE CTR=CTR+1 ::
TV(CTR)=VAL(TT$(K,2)):: J=J+LEN(TT$(K,1)
)-1 :: GOTO 240
200 IF FG=0 THEN CTR=CTR+3 :: TV(CTR-2)=
200 :: CW=CTR-1 :: NC=1 :: TV(CTR)=0 ::
FG=1 :: GOTO 240
210 CTR=CTR+1 :: TV(CTR)=Q :: NC=NC+1 ::
GOTO 240
220 NEXT K
230 PRINT "ERRORE NELLA FORMULA" :: FOR
TM=1 TO 500 :: NEXT TM :: PRINT :: GOTO
170
240 NEXT J :: IF FG=1 THEN TV(CW)=NC
250 CTR=CTR+1 :: TV(CTR)=0 :: N=CTR :: P
1=0 :: P2=0 :: FOR J=1 TO N :: P1=P1+(TV
(J)=182):: P2=P2+(TV(J)=183)
260 NEXT J :: IF P1<>P2 THEN 230 ELSE I=
E-1 :: BY=N+3 :: CALL LOAD(I,BY)
270 CTR=0 :: FOR I=E+3 TO E+2+N :: CTR=C
TR+1 :: BY=TV(CTR):: CALL LOAD(I,BY):: N
EXT I
280 INPUT "X = ":X :: PRINT "F(X) =":Y :
PRINT "STESSA FUNZIONE? (Y/N)"
290 CALL KEY(O,K,S):: IF S=0 THEN 290 EL
SE CALL HCHAR(23,3,32,28)
300 IF K=78 THEN 170 ELSE IF K=89 THEN 2
80 ELSE 290
310 DATA ")", "182", "(", "183", "=", "190", "
+", "193", "-", "194", "*", "195", "/", "196"
320 DATA "^", "197", "ATN", "204", "COS", "20
5", "INT", "207", "LOG", "208", "SIN", "210", "
TAN", "212"
330 DATA "PI", "221", "X", "88"
```

di Tullio PolICASTRO

Come realizzare la funzione Print Using sullo Spectrum

Nel ben noto computer della Sinclair, che pure dispone di molte caratteristiche interessanti (accresciute da quando si è resa disponibile l'Interfaccia 1), mancano alcune funzioni o comandi Basic che alcuni dei concorrenti invece hanno di normale dotazione.

Una di queste caratteristiche è quella di poter modellare le stampe di valori numerici secondo un formato prefissato: ciò è particolarmente utile quando si devono a esempio stampare tabelle in cui diversi importi debbono essere disposti opportunamente incolonnati (a esempio giustificati a sinistra sulla cifra delle unità, se interi; oppure con i punti decimali su di una stessa colonna, se comprendono una parte decimale).

Alcuni Basic dispongono a questo scopo del comando Print Using; a esempio dopo aver inserito Print Using ###.###, n il numero n verrà stampato con tre cifre intere (premettendo degli spazi vuoti se le cifre della parte intera sono meno di tre) e quattro cifre decimali (arrotondate se superassero tale numero, o completate con degli zeri se fossero in numero inferiore). Lo Spectrum non dispone di questa caratteristica: però non è difficile realizzarla, con una opportuna subroutine che può venire richiamata ogni volta che si devono stampare numeri secondo un determinato formato, qualcosa di strettamente analogo e perfettamente idoneo allo scopo.

La subroutine in questione è relativamente semplice, e occupa i numeri di linea da 10 a 30 del programmino riportato figura 1.

Prima della chiamata della routine devono venire definiti (una volta per tutte, oppure modificando valori precedenti se si desiderano adoperare formati diversi in varie occasioni) i valori delle variabili:

- x = numero di cifre intere (se il numero da stampare avrà un numero di cifre intere inferiore a x verranno aggiunti in testa

degli spazi, oppure un singolo o prima del punto decimale se il numero fosse inferiore a uno);

- y = numero di cifre decimali desiderate (il numero verrà stampato con l'ultima decimale arrotondata al valore superiore, se la cifra successiva è maggiore di cinque; se i decimali esistenti sono in numero inferiore a y, verranno aggiunti degli 0 non significativi).

Naturalmente prima della chiamata della routine bisognerà pure definire il valore di n, numero da stampare (nel programmino dimostrativo esso viene ricavato per Input).

Come dimostra la riga 100, a cui in definitiva si riduce il demo, si è previsto che nel caso che il numero n fosse superiore al massimo numero di x cifre intere, ossia la lunghezza della parte intera fosse superiore a x, non venga effettuata alcuna stampa, ma venga emesso un segnale di errore (acustico, nel caso), e ripetuta la richiesta del valore di Input. Questa regola può venire modificata, se così desiderato, adattandola alle proprie necessità.

La tecnica adoperata nella subroutine, come è abbastanza naturale, passa attraverso la conversione del valore numerico di n in una stringa, sulla quale poi si interviene grazie alle potenti istruzioni di slicing di cui è dotato lo Spectrum, facendo pure ricorso alle funzioni Str\$, Abs, Int e Len.

La subroutine utilizza alcune variabili interne (ni, s\$, m\$, d\$, d, l) per cui bisogna stare attenti a non usare gli stessi nomi di variabili nel programma principale (ovve-

ro, modificare i nomi usati nella subroutine). Il loro significato è il seguente:

- ni = parte intera di n (o meglio del suo valore assoluto, se n è negativo);
- n\$ = stringa corrispondente a ni;
- l = lunghezza della stringa n\$ (= numero di cifre di n intere; 1 comunque se n < 1);
- d = parte decimale di n.

m\$ e d\$ vengono costruite opportunamente dalla subroutine, tenendo conto dei casi particolari.

Si noti che, se il numero di cifre intere o decimali di n fosse superiore a otto o a dieci, rispettivamente, si dovrebbero incrementare i numeri degli spazi vuoti o degli 0 specificati nelle stringhe costanti, poste fra virgolette, delle linee 25 e 15 rispettivamente.

La subroutine è in grado di accettare anche numeri nel formato (per esempio) 1,2 E 3 (esponenziale), purché superino il test sul numero di cifre intere inserito nel programma (vedi linea 100).

Per un uso pratico, poiché la subroutine comprende una propria istruzione generica di Print che prescinde dall'indicazione della posizione iniziale, questa dovrà essere predefinita prima della chiamata della subroutine tramite un'istruzione tipo Print At r,c; (il punto-e-virgola finale è essenziale).

Poiché, come si è detto, con l'aggiunta di eventuali spazi la subroutine stampa sempre numeri col medesimo numero di cifre intere (comprendendo fra queste anche gli spazi), l'effetto è di un allineamento ordinato, se nelle righe successive la posizione

Figura 1 - La funzione Print Using.

```

5 INPUT x,y
6 GO TO 100
10 REM Subroutine PRINT USING
11 REM All input, x=n.o cifre
interi; y=n.o cifre decimali; n
= numero da stampare
12 LET s$="( " AND n>=0)+1"-"
AND n<0): LET n=ABS n
15 LET ni=INT n: LET n$=STR$ n
1: LET l=LEN n$: LET d=n-ni: IF
d<10+(-y) THEN LET d$=".00000000
00" ( TO y+1): GO TO 25
20 LET d$=(STR$ (d+5*10+(-y-1)
)) (1+(d>=.1) TO y+1+(d>=.1))
25 LET m$=" " ( TO x-1)+
n$
30 PRINT s$+m$+d$: RETURN
100 INPUT n: IF LEN STR$ INT AB
S n>x THEN BEEP .5,20: GO TO 100
110 GO SUB 10: GO TO 100
    
```

ZX SPECTRUM

(=colonna) iniziale di stampa viene mantenuta fissa. In altre parole, fissando la posizione iniziale col Print At di cui sopra (naturalmente è anche possibile stampare da inizio riga senza nessun Print preliminare), i numeri stampati su righe successive avranno il punto decimale allineato.

Una caratteristica molto importante di questo procedimento è che con esso è possibile stampare con lo Spectrum numeri con qualsiasi numero di cifre intere o decimali, superando il limite delle otto cifre significative posto dal normale Print di costanti o variabili. Si può inoltre ancora fare notare che se si pone y (numero di cifre decimali) eguale a 0, non vengono stampati decimali, ma permane la presenza del punto decimale dopo la cifra delle unità: se si vuole eliminare questo inconveniente occorre sostituire nella linea 15 al punto "." iniziale posto prima della serie di 0 uno spazio. Si deve però osservare che in questo particolare caso non si ha arrotondamento ma solo troncamento alla parte intera. ■

di Enrico Martelli

L'uso dell'unità a dischi Commodore

Questo programma si rivolge agli utilizzatori del Commodore C 64 e dell'unità dischi 1541. Quando si passa dall'uso del registratore a quella del disco, a volte, ci si trova spaesati dato il diverso trattamento da riservare a quest'ultima periferica. Facciamo un po' di ripasso dei comandi essenziali per il corretto funzionamento del sistema e per lo sfruttamento a pieno delle possibilità offerte dalla macchina.

Appena acquistato un dischetto la prima operazione da fare è la formattazione; essa consiste nel preparare il supporto per le successive operazioni di scrittura e lettura. Senza tale operazione il sistema operativo non riconosce come suo il disco e non è in grado di operare su di esso.

Per effettuare questa funzione bisogna, una volta inserito il dischetto nel drive, dare direttamente questi comandi: Open 15,8,15,"N:Nome disco, ID": Close

15 e <Ret>. Il nome del disco è una stringa alfanumerica massimo lunga 18 caratteri mentre l'ID sono due caratteri alfanumerici anch'essi che identificano il disco

COMMODORE 64

Figura 1 - Tabella di conversione.

{HOME}HOME
{CLR}PULIZIA SCHERMO
{CUR.SU}CURSORE IN ALTO
{CUR.GIU}CURSORE IN BASSO
{CUR.DES}CURSORE A DESTRA
{CUR.SIN}CURSORE A SINISTRA
{SPC}SPAZIO
{RVS ON}REVERSE ON
{RVS OFF}REVERSE OFF
{INST}INSERT
{F1}TASTO F1
{F2}TASTO F2
{F3}TASTO F3
{F4}TASTO F4
{F5}TASTO F5
{F6}TASTO F6
{F7}TASTO F7
{F8}TASTO F8
{BLACK}COL. NERO (CTRL+1)
{WHITE}COL. BIANCO (CTRL+2)
{RED}COL. ROSSO (CTRL+3)
{CYAN}COL. CIANO (CTRL+4)
{PURPLE}COL. PORPORA (CTRL+5)
{GREEN}COL. VERDE (CTRL+6)
{BLUE}COL. BLU (CTRL+7)
{YELLOW}COL. GIALLO (CTRL+8)
{ORANGE}COL. ARANCIO (CBM+1)
{BROWN}COL. MARRONE (CBM+2)
{LT.RED}COL. ROSSO CHIARO (CBM+3)
{GRAY1}COL. GRIGIO 1 (CBM+4)
{GRAY2}COL. GRIGIO 2 (CBM+5)
{LT.GREEN}COL. VERDE CHIARO (CBM+6)
{LT.BLUE}COL. BLU CHIARO (CBM+7)
{GRAY3}COL. GRIGIO 3 (CBM+8)

I CARATTERI GRAFICI, OTTENUTI CON LA PRESSIONE DEI TASTI 'SHIFT' E 'CBM', SONO CODIFICATI IN MODO DA INDICARE IL TASTO DA PREMERE ASSIEME A 'SHIFT' O 'CBM'. ES. IL CUORICINO E' CODIFICATO CON {SH S}.

UN NUMERO DENTRO LE PARENTESI INDICA LE VOLTE CHE IL TASTO VA PREMUTO.

stesso. Sarebbe opportuno che ogni dischetto ne avesse uno differente. La formattazione potrà essere anche parziale dando il comando nel seguente modo:

I SEGRETI DEI PERSONAL

Open 15,8,15,"N:Nome disco": Close 15; in questo modo esso manterrà lo stesso numero di ID e l'operazione sarà molto più rapida.

Altra procedura è quella di copia essa permette di copiare su uno stesso disco il medesimo file con un nome diverso. I comandi da dare sempre in modo diretto sono: Open 15,8,15,"C:Nome da assegnare alla copia=Nome file da copiare": Close 15; i nomi non debbono essere lunghi più di 16 caratteri.

Esiste anche la procedura di Rename utile quando si vuole cambiare il nome a un programma già precedentemente memorizzato, per fare ciò bisogna digitare: Open 15,8,15,"R:Nuovo nome=Vecchio nome":Close 15, anche in questo caso i due nomi non potranno essere più lunghi di 16 caratteri.

Un'operazione pericolosa, ma ugualmente utile, è quella chiamata Scratch: permette la cancellazione di un file presente sul dischetto. I comandi da dare sono: Open 15,8,15,"S:Nome del file da cancellare": Close 15 (il nome sempre 16 caratteri). Il file così cancellato non è più presente nella directory del dischetto e quindi non più caricabile; esso si potrebbe recuperare con altri programmi di utility. Un altro comando da dare in modo diretto è il Validate esso serve per cancellare file mal chiusi, recuperare spazio sul dischetto eventualmente mal sfruttato, eccetera. Per l'uso di questo comando bisogna stare attenti se sul dischetto sono presenti file del tipo "Rel" nel qual caso essi verranno cancellati, i comandi da dare sono: Open 15,8,15,"V": Close 15.

Uno degli aspetti più interessanti nell'uso dell'unità floppy è quello di avere la possibilità di leggerne il contenuto in una specie di indice chiamato directory; su di esso sono riportati i seguenti dati: nome del disco, identificazione dello stesso, formato Dos (A2), i nomi dei programmi memorizzati (il numero di blocchi occupati da essi e il tipo: Prg, Seq, Rel, Usr). Il numero di blocchi liberi. Ogni blocco contiene 256 byte. Per ottenere la directory bisogna digitare: Load "\$", 8 <Ret> e poi <List> essa naturalmente potrà essere stampata su carta usando questi comandi: Open 4,4: Cmd 4: List e, per chiudere il canale con la stampante Print#4:Close 4. Purtroppo il Basic limitato implementato sul C 64 fa sì che ogni volta che si chiedi la directory viene cancellato il programma Basic in memoria (non esiste il comando Catalog!).

Dopo questo lungo ripasso veniamo a quello che il programma intende offrire. Esso vorrebbe essere un valido supporto per l'uso dei comandi testè analizzati. Una volta digitato il programma, prevede, con l'aiuto di un semplice menu, tutte le varie opzioni di cui sopra. Scelta l'opzione desiderata, confermata l'intenzione di proseguire, il programma vi guida all'inserimento di tutti i dati necessari affinché questa vada a buon fine. Esiste sempre la possibilità di ritornare al menu principale semplicemente premendo la "M". Una funzione di Input controllato permette la verifica costante di tutti i dati immessi.

Vediamo ora le principali caratteristiche del programma. Come potrete notare il menu si presenta in una veste insolita rispetto a quella che siamo normalmente

abituati a vedere. L'implementazione di esso va dalla riga 20 alla riga 280 (esclusa la riga 30) e può con facilità essere inglobato in un qualsiasi vostro programma. Per modificare il numero delle opzioni basta cambiare il valore della variabile N alla linea 40. Dimenticavamo: questo menu riesuma un carattere raramente usato il Chr\$(186) che nel set minuscolo appare così (✓).

Si può vedere anche che dopo ogni Input non appare sullo schermo l'antiestetico punto interrogativo, per fare ciò si è ricorso a un piccolo artificio: lo si è fatto diventare dello stesso colore dello sfondo (vedi le istruzioni delle righe 330-360-480 eccetera).

Interessante anche la routine che va dalla riga 1000 alla riga 1190 che permette di leggere la directory senza uscire dal programma stesso e, molto più importante, senza cancellarlo; anche questa routine potrà essere implementata e usata nei vostri programmi.

Il programma prevede anche una routine di intercettazione e segnalazione di errori provenienti dal non corretto uso dell'unità 1541. (Vedere istruzioni alle righe 1290/1350). Ultima cosa, con l'istruzione che appare alla riga 10 si inibisce la funzione di Run/Stop Restore e quindi per uscire dal programma, bisogna optare per la scelta numero sette che riporta tutto alle condizioni normali di lavoro. Un consiglio: inserite questa istruzione solo quando siete sicuri che il programma giri perfettamente. ■

Listato 1 - Il programma per l'uso dell'unità a dischi Commodore.

```
10 POKE808,225
20 PRINTCHR$(14)CHR$(8):POKE53280,246
30 DE$="{CUR.SU}{33 SPC}{CUR.SU}"
40 N=7
50 CR$="{RVS ON}{2 SPC}"+CHR$(186)+"{3 SPC}{RVS ON}"+"{5 CUR.SIN}"
60 E$="{5 SPC}{5 CUR.SIN}"
70 PRINT"{CLR}";
80 PRINT"{RVS ON}{8 SPC}* {SH D}OS. {SH E}. {SH M}ARTELLI *{9 SPC}{RVS ON}"
90 PRINTTAB(10)"{CUR.GIU}1) {SH F}ORMATT AZIONE"
100 PRINTTAB(10)"2) {SH C}OPIA"
110 PRINTTAB(10)"3) {SH R}ENAME"
120 PRINTTAB(10)"4) {SH S}CRATCH"
```

```
130 PRINTTAB(10)"5) {SH V}ALIDATE"
140 PRINTTAB(10)"6) {SH D}IRETTORIO"
150 PRINTTAB(10)"7) {SH F}INE OPERAZIONI"
160 PRINT:PRINT:PRINT:PRINT
170 PRINT"{5 SPC}{RVS ON}{SH P}REMERE SPAZIO PER SELEZIONARE"
180 PRINT"{5 SPC}{RVS ON}{30 SPC}"
190 PRINT"{5 SPC}{RVS ON}{SH P}REMERE RETURN PER LA CONFERMA"
200 PRINT"{HOME}{2 CUR.GIU}";:L=1
210 PRINTCR$
220 GETA$:IFA$=""THEN220
230 IFA$<>CHR$(13) AND A$<>CHR$(32)THEN220
240 IFA$=CHR$(13)THEN270
250 PRINTE$:L=L+1:IFL>NTHEN200
260 GOTO210
```

COMMODORE 64

Seguito listato 1.

```

270 PRINT"{HOME}";:FORI=1TON+8:PRINT:NEX
T
280 ON L GOTO290,440,590,740,900,980,120
0
290 REM
300 XX$="{SH F}ORMATTAZIONE":GOSUB1230
310 PRINT"{CLR}{RVS ON}{11 SPC}* {SH F}O
RMATTAZIONE *{10 SPC}{RVS ON}"
320 PRINT"{3 CUR.GIU}{2 SPC}NOME DEL DIS
CO (MAX 18):"
330 POKE631,154:POKE198,1:INPUT"{BLUE}";
ND$
340 IFLEN(ND$)>18THENPRINTDE$:GOTO330
350 PRINT"{2 CUR.GIU}{2 SPC}{SH I}DENTIF
ICAZIONE (2 CAR):"
360 POKE631,154:POKE198,1:INPUT"{BLUE}";
ID$
370 IFLEN(ID$)>2ORLEN(ID$)=1THENPRINTDE$
:GOTO360
380 PRINT"{2 CUR.GIU}{2 SPC}{SH C}ONFERM
I S/N ({SH M} PER MENU)"
390 GETA$:IFA$=""THEN390
400 IFA$<>"S"ANDA$<>"N"ANDA$<>"M"THEN390
410 IFA$="N"THEN310
420 IFA$="M"THEN20
430 OPEN15,8,15:PRINT#15,"N:"+ND$+" "+ID
$:GOSUB1290:CLOSE15:GOTO20
440 REM
450 XX$="{SH C}OPIA":GOSUB1230
460 PRINT"{CLR}{RVS ON}{14 SPC}* {SH C}O
PIA *{14 SPC}{RVS ON}"
470 PRINT"{2 CUR.GIU}{2 SPC}{SH N}OME NU
OVO FILE (MAX 16):"
480 POKE631,154:POKE198,1:INPUT"{BLUE}";
NF$
490 IFLEN(NF$)>16THENPRINTDE$:GOTO480
500 PRINT"{2 CUR.GIU}{2 SPC}{SH N}OME FI
LE DA COPIARE (MAX 16):"
510 POKE631,154:POKE198,1:INPUT"{BLUE}";
FC$
520 IFLEN(FC$)>16THENPRINTDE$:GOTO510
530 PRINT"{2 CUR.GIU}{2 SPC}{SH C}ONFERM
I S/N ({SH M} PER MENU)"
540 GETA$:IFA$=""THEN540
550 IFA$<>"S"ANDA$<>"N"ANDA$<>"M"THEN540
560 IFA$="N"THEN460
570 IFA$="M"THEN20
580 OPEN15,8,15:PRINT#15,"C:"+NF$+" "+FC
$:GOSUB1290:CLOSE15:GOTO20
590 REM
600 XX$="{SH R}ENAME":GOSUB1230
610 PRINT"{CLR}{RVS ON}{14 SPC}* {SH R}E
NAME *{14 SPC}{RVS ON}"
620 PRINT"{3 CUR.GIU}{2 SPC}{SH N}UOVO N
OME PROGRAMMA (MAX 16):"
630 POKE631,154:POKE198,1:INPUT"{BLUE}";
NP$
640 IFLEN(NP$)>16THENPRINTDE$:GOTO630
650 PRINT"{2 CUR.GIU}{2 SPC}{SH N}OME VE
CCHIO PROGRAMMA (MAX 16):"
660 POKE631,154:POKE198,1:INPUT"{BLUE}";
VP$
670 IFLEN(VP$)>16THENPRINTDE$:GOTO660
680 PRINT"{2 CUR.GIU}{2 SPC}{SH C}ONFERM
A{2 SPC}S/N (M PER MENU)"
690 GETA$:IFA$=""THEN690
700 IFA$<>"S"ANDA$<>"N"ANDA$<>"M"THEN690
710 IFA$="N"THEN610
720 IFA$="M"THEN20
730 OPEN15,8,15:PRINT#15,"R:"+NP$+" "+VP
$:GOSUB1290:CLOSE15:GOTO20
740 REM
750 XX$="{SH S}CRATCH":GOSUB1230
760 PRINT"{CLR}{RVS ON}{14 SPC}* {SH S}C
RACTH *{14 SPC}{RVS ON}"
770 PRINT"{2 CUR.GIU}{2 SPC}{RVS ON}{SH
R}ICHIESTA ULTERIORE CONFERMA !!!{RV
S ON}":PRINT"{CUR.GIU}{2 SPC}{SH C}O
NFERMI S/N"
780 GETA$:IFA$=""THEN780
790 IFA$<>"S"ANDA$<>"N"THEN780
800 IFA$="M"THEN20
810 PRINT"{3 CUR.GIU}{2 SPC}{SH N}OME DE
L FILE DA CANCELLARE (MAX 16):"
820 POKE631,154:POKE198,1:INPUT"{BLUE}";
FL$
830 IFLEN(FL$)>16THENPRINTDE$:GOTO820
840 PRINT"{2 CUR.GIU}{SH E}' QUESTO IL
FILE DASA CANCELLARE S/N":PRINT"{2 S
PC}{SH M} PER MENU)"
850 GETA$:IFA$=""THEN850
860 IFA$<>"S"ANDA$<>"N"ANDA$<>"M"THEN850
870 IFA$="N"THEN760
880 IFA$="M"THEN20
890 OPEN15,8,15:PRINT#15,"S:"+FL$:GOSUB1
290:CLOSE15:GOTO20
900 REM
910 XX$="{SH V}ALIDATE":GOSUB1230
920 PRINT"{CLR}{RVS ON}{12 SPC}* {SH V}A
LIDATE *{14 SPC}{RVS ON}"
930 PRINT"{3 CUR.GIU}{2 SPC}{RVS ON}{SH
A}{2 SH T}{SH E}{SH N}{SH Z}{SH I}{S
H O}{SH N}{SH E}{RVS ON} AI FILE {SH
R}{SH E}{SH L}{SH A}{SH T}{SH I}{SH
V}{SH I} !!!":PRINT"{2 CUR.GIU} CON
FERMI S/N"
940 GETA$:IFA$=""THEN940
950 IFA$<>"S"ANDA$<>"N"THEN940
960 IFA$="M"THEN20
970 OPEN15,8,15:PRINT#15,"V":GOSUB1290:C
LOSE15:GOTO20
980 REM
990 XX$="{SH D}IRETTORIO":GOSUB1230
1000 PRINT"{CLR}{RVS ON}{11 SPC}* {SH D}
IRETTORIO *{14 SPC}{RVS ON}"
1010 OPEN15,1,15:OPEN1,8,0,"$":GOSUB1290
:IFB$<>"OK"THENCLOSE1:GOTO20
1020 GET#1,A$,B$
1030 GET#1,A$,B$
1040 GET#1,A$,B$
1050 C=0:IFA$<>" "THENC=ASC(A$)
1060 IFB$<>" "THENC=C+ASC(B$)*256
1070 PRINTMID$(STR$(C),2;TAB(4);

```

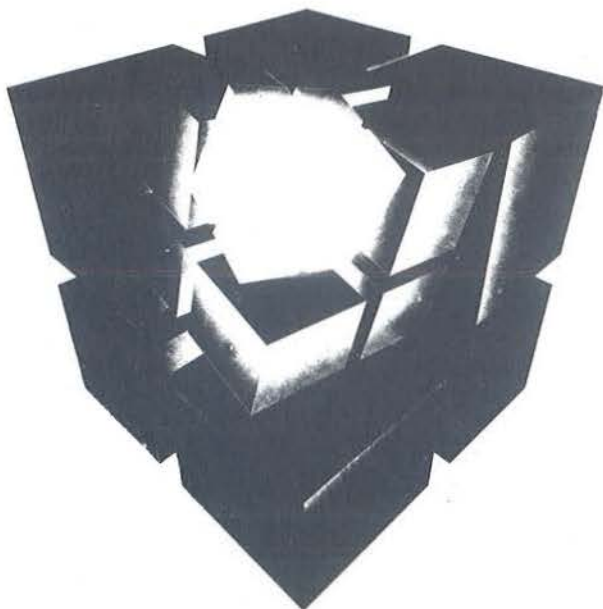
Seguito listato 1.

```

1080 GET#1,B$:IFST<>0THEN1160
1090 IFB$<>CHR$(34)THEN1080
1100 GET#1,B$:IFB$<>CHR$(34)THENPRINTTB$
;:GOTO1100
1110 GET#1,B$:IFB$=CHR$(32)THEN1110
1120 PRINTTAB(22);:C$=""
1130 C$=C$+B$:GET#1,B$:IFB$<>""THEN1130
1140 PRINTLEFT$(C$,3)
1150 IFST=0THEN1030
1160 PRINT"BLOCCHI LIBERI"
1170 CLOSE1:CLOSE15:PRINT:PRINT"{RVS ON}
PREMI UN TASTO{2 SPC}QUALUNQUE PER
IL MENU.{2 SPC}"
1180 GETX$:IFX$=""THEN1180
1190 GOTO20
1200 REM
1210 XX$="{SH F}INE {SH O}PER.":GOSUB123
    
```

```

0
1220 PRINT"{CLR}":PRINTCHR$(142)CHR$(9):
POKE808,237:POKE53280,254:END
1230 PRINT"{CLR}{RVS ON}{7 SPC}* {SH R}I
CHIESTA CONFERMA *{10 SPC}{RVS ON}"
1240 PRINT"{HOME}{11 CUR.GIU}{SH C}ONFER
MI L'OPZIONE DI "XX$" S/N"
1250 GETA$:IFA$=""THEN1250
1260 IFA$<>"S"ANDA$<>"N"THEN1250
1270 IFA$="N"THEN20
1280 IFA$="S"THENRETURN
1290 INPUT#15,A,B$,C,D
1300 IFA$<20THENRETURN
1310 PRINT"{2 SPC}"A,B$,C,D
1320 CLOSE15
1330 PRINT"{4 CUR.GIU} {SH P}REMI UN TAS
TO"
1340 GETA$:IFA$=""THEN1340
1350 RETURN
    
```



SECONDO FORUM

CONFERENZE/ESPOSIZIONI

IBM PC

E COMPATIBILI

19/20/21 MARZO 1986
IL GIRASOLE - CENTRO PER IL
COMMERCIO INTERNAZIONALE
MILANO LACCHIARELLA

INFORMAZIONI: CAPRIC ORGANISATION ITALIA S.r.l. - VIA KORISTKA, 3 - 20154 MILANO - TEL. (02) 3490842

SERVIZIO SOFTWARE

Bit

Bit e Personal Software propongono ai propri lettori i dischi o le cassette dei programmi pubblicati. Uso e descrizione dei programmi si trovano sui rispettivi numeri delle riviste.

PERSONAL SOFTWARE



Bit n°	Programma	Sistema	Codice	Supporto	Bit n°	Programma	Sistema	Codice	Supporto
38	Gioco della scimmia	Vic 20	V1381	Cassetta	67	Videoqust 85	Max	MS671	Cassetta
	Spacemattioni				68	Fianoti e microcomputer	Apple II	AP682	Disco
38	Pianet	Apple II	AP382	Disco	68	Un computer per la didattica	C 16	C1681	Cassetta
42	Apple-Chef	Apple II	AP422	Disco	68	Classifica Campionato di calcio	Max	MS681	Cassetta
42	Provariflessi	Vic 20	V1421	Cassetta	69	Grafici, diagrammi istogrammi	Apple II	AP692	Disco
45	Tiny Forth	Apple II	AP452	Disco	69	Calcolatrice in Rpn	C 16	C1691	Cassetta
45	Ali Babà	ZX Spectrum	SP451	Cassetta	69	Calorimetro	C 64	C6692	Disco
46	Forzaquattro	Apple II	AP462	Disco	69	Classifica Campionato F 1	Max	MS691	Cassetta
46	Simulavolo	ZX Spectrum	SP461	Cassetta	69	Educazione musicale	T199/4A	T1691A	Cassetta
48	Memory Alla IV	C 64	C6461	Cassetta	69	Billard Pocket	T199/4A	T1691B	Cassetta
49	Scorpion	Apple II	AP492	Disco	70	Set di caratteri	C 16	C1701	Cassetta
50	Fp-Fict	Apple II	AP502	Disco	70	Pitoner 2.0	M20	M2702	Disco
50	Prima e Terza	ZX Spectrum	SP501	Cassetta					
51	Magicalog	Apple II	AP512	Disco					
53	Partita a golf	Vic 20	V1531	Cassetta					
53	Analisi numerica	C 64	C6531	Cassetta					
53	PI/Bit: il compilatore	Apple II	AP532	Disco	3	La carta del cielo. Collisione	Apple II	AP032	Disco
55	Costellazioni	Apple II	AP542	Disco	4	Interi in precisione multipla			
55	Come polarizzare i transistor	C 64	C6541	Cassetta					
58	Memory Omega I	C 64	C6582A	Disco	5	Pretty printer, Shape table	Apple II	AP052	Disco
58	Copy disk per C 64	C 64	C6582B	Disco	7	Data base modulare	Apple II	AP072	Disco
59	Checksum 64	C 64	C6592	Disco	14	Tool-Kit	C 64	C6141	Cassetta
59	Checksum 64	C 64	C6591	Cassetta	19	Type-Writer	Vic 20	V1192	Disco
59	Data-Bank	ZX Spectrum	SP591	Cassetta	20	Scopa	C 64	C6201	Cassetta
60	Life Hgr	Apple II	AP602	Disco	30	Cero-Race	ZX Spectrum	SP601	Cassetta
60	Tutti pittori	C 64	C6601	Disco	31	Progetto aeromodello	MAX	MS311	Cassetta
60	Difesa della Terra	C 16	C1601	Cassetta	31	Scopa a tre carte	Apple II	AP312	Disco
60	Loat on the pack	Sega	SE601	Cassetta	31	Grand Prix	C 64	C6312	Disco
60	Seidata e Wordproc	ZX Spectrum	SP601	Cassetta	31	Sprite	ZX Spectrum	SP311	Cassetta
60	Il Barone Rosso	T199/4A	T1601	Cassetta	31	Box	C 16	C1311	Cassetta
60	Word processor	C 64	C6602	Disco	32	Animazioni di funzioni			
60	Othello	Vic 20	V1601	Cassetta	32	Animazioni di funzioni	Zx Spectrum	SP321	Cassetta
61	Disegno di mobili componibili	C 16	C1611	Cassetta	32	Prospettiva e grafica tridimensionale	C 64	C6321	Cassetta
61	Esperimento di Milikan	C 64	C6611	Cassetta	33	Music editor	C 64	C6331	Cassetta
61	Esperimento di Milikan	C 64	C6612	Disco	33	Bogey Boogie	T199/4A	T1331	Cassetta
61	Disegnare in alta risoluzione	ZX Spectrum	SP611	Cassetta	33	Totocalcio	Vic 20	V1331	Cassetta
61	Printing music	T199/4A	T1611	Cassetta	34	Progettazione filtri attivi	Sharp MZ-700	SH331	Cassetta
61	Musica facile	Sega	SE611	Cassetta	34	Sprite per disegnare	ZX Spectrum	SP341	Cassetta
61	Vic-Calc	Vic 20	V1611	Cassetta	34	Battaglia galattica	Apple II	AP342	Disco
62	Gestione conto corrente	C 64	C6622	Disco	35	Briscola	Sharp MZ-700	SH351	Cassetta
62	Gioco della tombola	T199/4A	T1621	Cassetta	35	Foras 4	T199/4A	T1351	Cassetta
62	Aspc: lo Spectrum contro la carie	ZX Spectrum	SP621	Cassetta	35	Totocalcio	Max	MS351	Cassetta
63	Operazioni in pagina grafica	Apple II	AP632	Disco	35	Mincitor 64	C 64	C6351	Cassetta
63	Grafici a barre verticali	C 16	C1631	Cassetta	35	Tappeto mobile	Apple II	AP352	Disco
63	Contabilità casalinga	C 64	C6632	Disco	36	Fuga al buio	Apple II	AP362A	Disco
63	Disegnare sullo schermo	Max	MS631	Cassetta	36	Pro dotto	Apple II	AP362B	Disco
63	Over Basic	ZX Spectrum	SP631	Cassetta	36	M20 Paint	M20	OL362	Disco
63	Le otto Regine	T199/4A	T1631	Cassetta	36	Il campanaro	Apple II	AP362C	Disco
63	Egitto 2000	Vic 20	V1631	Cassetta	36	Lucy writer	ZX Spectrum	SP361	Cassetta
64	Roller	Apple II	AP642	Disco	36	Free sky	Sharp MZ-700	SH361	Cassetta
64	Come sistemiamo i mobili?	C 16	C1641	Cassetta	36	Chess training	T199/4A	T1361	Cassetta
64	Titolazioni simulate	C 64	C6642	Disco	36	Input simulato	Max	MS361	Cassetta
64	Defektor	Max	MS641	Cassetta	37	Grafica 3D	C 16	C1371	Cassetta
64	Color composer	Sharp MZ-700	SH641	Cassetta	37	Panico	Zx Spectrum	SP371	Cassetta
64	Computer music	T199/4A	T1641	Cassetta	37	Cruci Max	Max	MS371	Cassetta
65	Integer compiler	ZX Spectrum	SP651	Cassetta					
66	Battaglia navale	C 16	C1661	Cassetta					
67	Analisi di reti elettriche	Apple II	AP672	Disco					
67	Concerto grosso	C 16	C1671	Cassetta					
67	Gestione contabilità magazzino	C 64	C6672	Disco					

Tutti i dischi e le cassette dei programmi sono in vendita a L. 15.000 ciascuno.

Per richiedere i programmi in contrassegno, pagando direttamente al postino la cifra indicata, inviare il seguente tagliando
Spedire in busta chiusa a Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

Inviatemi i seguenti nastri e/o dischi con i programmi pubblicati su Bit - Personal Software

Cod. _____ a L. **15000**

Cod. _____ a L. _____

Cod. _____ a L. _____

Cod. _____ a L. _____

+ SPESE POSTALI
(contributo fisso)

3000

TOTALE L. _____

che pagherò al postino alla consegna del pacco



GRUPPO EDITORIALE JACKSON

Cognome.....

Nome.....

Indirizzo.....

CAP.....

Città.....

Firma.....

CON INSERTO SUPERBIT
64 PAGINE DI SOFTWARE PER IL TUO PERSONAL

è in edicola il nuovo numero

*Bit, la prima
e più diffusa rivista
di personal computer
e accessori*



**UNA
PUBBLICAZIONE
DEL
GRUPPO
EDITORIALE
JACKSON**

SAN FRANCISCO-LONDRA-MILANO



Bitest
Philips :Yes

Morrow
Pivot II Elite

Cartridge
Date - time C 64

Mar modem

Bstep per Apple

Pixel per Msx

Interpolazione
per QL

Lancio
col paracadute
per Atari

Set di caratteri
per C 16

Plotter 2.0
per M20

**SPECIALE:
INTELLIGENZA
ARTIFICIALE**

PERSONAL MARKET

Apple

Cambio software per Apple senza scopo di lucro. Dispongo di molti programmi; assicuro la massima serietà. Scrivere a: Massimo Giladetti - Via Borsi, 62 - 28100 Novara

Vendo **Apple II plus 48 Kbyte**, monitor a fosfori verdi, drive con interfaccia, stampante bidirezionale 80-132 colonne con interfaccia, dischi; il tutto a L. 2.000.000. Sergio Carone - Via Bagnana, 13/A - 52100 Arezzo - Tel. 0575/966043

Vorrei contattare utenti di **Macintosh - Apple II - Ibm** per scambi di software ed esperienze. Mauro Tazzari - Via Canalazzo, 44 - 48100 Ravenna - Tel. 0544/461687

Cambio software per **Macintosh**. Dispongo di molti programmi. Giovanni Meloni - Via L. Canali, 6 - 02100 Rieti - Tel. 0746/45134

Desidero iniziare con tutti gli applisti italiani uno scambio di programmi per **Apple IIe**. Massima serietà, contattare: Giovanni Gardini - Via F. Coradini, 2 - 52100 Arezzo

Per **Apple II - II plus** vendo **drive 5"** con alimentatore, piattina per 2 drive, controller per interfacciare tutti i drive a 40 tracce; il tutto a L. 350.000. Vendo inoltre schede Nuova Elettronica e Micro Design. Ottimi prezzi. Telefonare ore serali. Massimo Nonato - Via XX Settembre, 32 - 28010 Gargallo (NO) - Tel. 0322/95504

Commodore

Vendo **Vic 20**, un anno di vita, più registratore 1530 ed espansione 16 Kbyte a L. 200.000. Giovanni De Virgillis - Via Europa, 30 - 20097 S. Donato Milanese (MI) - Tel. 02-5272109

Vendo **C 64 più registratore** e 200 programmi su cassetta (giochi, utility e applicativi) a L. 350.000. Solo 6 mesi di vita, imballi originali, spese a mio carico. Emidio Borzi - Via Cellini, 44 - 74029 Talsano (TA) - Tel. 099/511761

Vendo **C 64 più registratore C2N**, numerosissimi giochi e utility con relativi manuali, Quickshot II più tasto reset a sole L. 320.000. Vi aspetto, telefonate! Fabrizio - Milano - Tel. 02/5275578

Vendo **Commodore Pet 2001** più 115 giochi e programmi, monitor, registratore, cassetta pulisci testina a L. 550.000. Telefonare ore pasti o alle 18.00 oppure scrivere. Fabio Talamini - Via Eroi del Lavoro, 12 - 04100 Latina - Tel. 0773/498036

L'Eporedia computer club cede **cartridge Fast Driver** per C 64 ed espansione 3-8-16 Kbyte per Vic 20. Graditi scambi di programmi per C 64. Eporedia Computer Club - c/o Carlo Laiolo - Via Rovagnone, 14 - 10015 Samone (TO) - Tel. 0125/53115

Per C 64 compro **giochi su nastro o, preferibilmente, su disco**. Cerco Pac man, Mrs. Pac man, Pitfall e Ghostbusters a prezzi modici. Walter Raffaelli - Via Mazzini, 125 - 26013 Crema

Vendo **C 64 più registratore** a sole L. 450.000, in omaggio cassette con programmi di giochi e utility. Assicuro e richiedo la massima serietà. Scrivere o telefonare a: Antonio Malozzi - Via Privata Ciuffo, 6 - 04020 M. di Minturno (LT) - Tel. 680016

Sinclair

Vendo **Zx Spectrum Plus con interfaccia programmabile**, joystick, 40 programmi, libro "Alla scoperta dello Zx Spectrum" a L. 400.000 trattabili. Solo zona Venezia. Alessandro Paoletti - Via Crea, 75/F - 30038 Spinea (VE) - Tel. 041/996301

Il Club Spectromania offre **12 programmi fra i migliori** per il tuo Zx Spectrum: Hyper sports, Superboxe challenge, ecc. Tutto compreso a sole L. 30.000. Club Spectromania - Via delle Romite, 8 - 50124 Galluzzo (FI) - Tel. 055/2048705

Compro **Zx Spectrum normal 48 Kbyte**. Telefonare dal lunedì al venerdì ore pasti allo 0545/27637 oppure sabato e domenica allo 0541/34835 chiedendo di Mauro. Mauro Ferri - Via Stoccolma, 29 - 47037 Rimini (FO) - Tel. 0541/34835

Vendo in blocco o separatamente oltre 1.000 programmi **Spectrum** al miglior offerente (min. L. 25.000). Vendo inoltre 50 programmi **Mx** a L. 100.000. Dorian Orlandi - Via Quasimodo, 6/C - 46023 Gonzaga (MN) - Tel. 0376/588555

Vendo **Spectrum Plus con oltre 1.000 programmi** al miglior offerente (min. L. 500.000). Vendo anche separatamente. Annuncio sempre valido. Andrea Cicogna - Via Quasimodo, 6/C - 46023 Gonzaga (MN) - Tel. 0376/588555

Vendo **Zx81 perfetti con cavetti**, alimentatore, manuale della casa, 3 libri (valore L. 48.000) e due giochi, Centipede e Star trek a L. 85.000 trattabili. Andrea Focardi - Via G. Di Vittorio, 56 - 50015 Grassano (FI) - Tel. 055/64204

Texas

Per **Ti99/4A** vendo **disk drive** a L. 200.000. Telefonare ore sbrali. Lello Zorzato - Via Fedro, 4 - 80126 Napoli - Tel. 081/7427068

Vendo **Ti99/4A più Extended Basic**, manuali di: introduzione al Basic, giochi ed Editor/Assembler. Tutto come nuovo a L. 300.000. Telefonare ore 20.00. Sergio Nova - Via della Torre, 31 - 20127 Milano - Tel. 02/2895347

Favoloso! Vendo Per **Ti99/4A Editor/Assembler più Multipian completo di manuale** e relativi dischetti, minimemory, Terminal emulator II, Statistics, Moonmine, Othello, Munchman e Tombstone city. Tutto il materiale è nuovissimo, ancora imballato; L. 999.000. All'acquirente regalo 23 dischetti più cassette con 500 programmi professionali e interfaccia per disk drive. Rocco Macri - c/o Istituto Serafico - 06081 Assisi (PG) - Tel. 075/812546

Vendo rivista con cassetta **Ti99 New Soft dal nr. 0 al nr. 12** a L. 70.000 in blocco o a L. 8.000 cadauna; stampante Seikosha Gp50A interfaccia Centronics a L. 230.000. Giorgio Castagnaro - Via Buonarroti, 13 - 21013 Gallarate (VA) - Tel. 0331/796139

Svendo **modulo Extended Basic** e cassette giochi per Texas **Ti99/4A**. Telefonare a: Ercole Donati - Via Appennini, 13 - 20151 Milano - Tel. 02/3533564

Vendo **Ti99/4A più registratore originale**, modulatore, alimentatore, interfaccia registratore, coppia joystick, Extended Basic, manuali, Sss Attack, cassette programmi e riviste specifiche a L. 400.000. Fabio Cerallo - Viale di Trastevere, 70 - 00153 Roma - Tel. 06/581607

Varie

Cambio programmi per sistema **Mx**. Telefonare ore pasti. Andrea Pimpolari - Via R. Malatesta, 58 - 00176 Roma - Tel. 06/294106

Vendo computer **Mx Philips Vg-8010** a L. 400.000 trattabili. Fedele Schipani - Via N. Sauro, 2 - 84091 Battipaglia (SA) - Tel. 0828/23633

Vendo **Ibm Pc 256 Kbyte Ram, due floppy disk da 360 Kbyte** più stampante grafica Ibm 80 colonne. Usato pochissimo, ancora in garanzia. Franco Maggioni - Via IV Novembre, 6 - 22067 Missaglia (CO) - Tel. 039/9200784

Cerco per **Mz-700 stampante Centronics GIp** in buone condizioni e a prezzo ragionevole. Telefonare a: Francesco Crispo - Via P. Amedeo, 221 B/4 - 00100 Roma - Tel. 06/7314982

Vendo **Atari 130Xe (due mesi di vita)** più registratore Atari, 5 cassette, 2 manuali, 1 joystick; il tutto, ancora in imballi originali, a sole L. 495.000. Maurizio Cagnesi - Via dei Cipressi, 6 - 10090 Bruino (TO) - Tel. 011/9086126

Soft Sharp club Roma cambia programmi per Mz-700. Tutti i programmi sono garantiti funzionanti. Per informazioni rivolgersi a: Francesco Loriga - Via Fulda, 115 - 00148 Roma - Tel. 06/5222504

Vendo per **Ibm Pc** e compatibili sistema operativo, **Basic, Turbo Pascal, Cobol**, alcuni copiatori. Prezzo da concordare. Per informazioni rivolgersi a: Andrea Maggi - Via F.lli Cervi, 8/D - 20010 Cornaredo (MI) - Tel. 02/9317678

Vendo computer **Ibm compatibile con 256 Kbyte di memoria**, video e doppio floppy, completo di programmi gestionali e programmi personalizzati. Prezzo affare. Claudio Giovanelli - Via Ripamonti, 194 - 20141 Milano - Tel. 02/53692

PERSONAL PICCOLLI SOFTWARE ANNUNCI

Sei un lettore di Personal Software e vuoi entrare in contatto con tutti gli altri lettori per comprare, cambiare o vendere il tuo sistema o qualche particolare accessorio? Spedisci questo tagliando a:
Gruppo Editoriale Jackson - Via Rosellini, 12 - 20124 Milano

COMPRO CAMBIO VENDO

Attenzione: Piccoli Annunci è aperta solamente alla vendita, l'acquisto o lo scambio di materiale fra privati, in un unico esemplare. Annunci di diversa natura verranno destinati.

Annuncio gratuito Annuncio evidenziato (allegare L. 5.000). Non si accettano fotocopie o fogli volanti allegati al coupon.

Nome Cognome

Via N. C.A.P.

Città Prov. Tel.

PERSONAL SOFTWARE LIBRI

Vuoi ordinare dei libri? Spedisci questo tagliando a:
Gruppo Editoriale Jackson
Via Rosellini, 12 - 20124 Milano

Nome Cognome _____

Indirizzo _____ N. _____

Cap. _____ Città _____ Provincia _____

Partita I.V.A. (Indispensabile per le aziende) _____

Si richiede l'emissione della fattura

Inviatemi i seguenti libri:					
Codice Libro	Quantità	Codice Libro	Quantità	Codice Libro	Quantità

Pagherò al postino il prezzo indicato + L. 3.000 per contributo fisso spese di spedizione.

Allego assegno n° di L.

Non abbonato Abbonato sconto Elettronica Oggi Automazione Oggi Elektor Informatica Oggi

Computerworld Bit Personal Software Strumenti Musicali Video Giochi Home computer Soft Dealer

Telecomunicazioni Oggi PC Magazine Personal O Compuscuola Super VIC Super Commodore

Data Firma

UNA PUBBLICAZIONE DEL GRUPPO EDITORIALE JACKSON

PERSONAL SOFTWARE

DIRETTORE RESPONSABILE: Giampietro Zanga

CAPO REDATTORE: Lorenzo Barriè

COORDINAMENTO TECNICO: Luca Zaninello

GRAFICA E IMPAGINAZIONE: Gabriele Cappuccio

SEGRETARIA DI REDAZIONE: Silvia Borgonovo

UFFICIO ABBONAMENTI: Tel. (02) 6880951/2/3/4/5

HANNO COLLABORATO A QUESTO NUMERO:

L. Accomazzi, V. Anselmo, M. Spero,
G. Tisi, L. Cassa, L. Zaninello,
U. Rossini, T. Palermo, S. Borsani,
T. Policastro, E. Martelli, M. Giacobazzi

CONSOciate ESTERE:

USA: G.E.J. PUBLISHING Group, Inc. a Subsidiary of Gruppo Editoriale Jackson - 1307 S. Mary Avenue - 94087 Sunnyvale (CA) - Tel. (408) 7730103 - Telex: 49959972

GRAN BRETAGNA: G.E.J. Publishing L.t.d. - 187 Oxford Street - London W1R1AJ
Tel. (01) 4392931 - Telex: (051) 21248

SPAGNA: Jackson Hispana S.A. - Plaza Republica del Ecuador, 2
28016 Madrid - Spagna - Tel. 1-2505820 - Telex: 052-49371 ELOCE

PUBLICITÀ: Concessionario per l'Italia e l'Estero
J. Advertising S.r.l. - Via Restelli, 5 - 20124 Milano
Tel. (02) 6882695-6882458-6880606 - Telex: 316213 REINA I

PUBLICITÀ PER USA E CANADA:
International Media Marketing - 16704 Marquardt Avenue,
P.O. Box 1217 Cerritos (CA) 90701 - (213) 926-9552

AGENTE PER LE TRE VENEZIE: Pierfilippo Obber - Via Nazionale, 70
38050 Imer (TN) - Tel. (0439) 67221.

FOTOCOPOSIZIONE: Lineacomp S.r.l. - Via Rosellini, 12 - 20124 Milano

STAMPA: Litografica S.r.l. - (Busto Arsizio - MI)

CONCESSIONARIO ESCLUSIVO PER LA DIFFUSIONE IN ITALIA E ALL'ESTERO: SODIP - Via Zuretti, 25 - 20125 Milano

AUTORIZZAZIONE ALLA PUBBLICAZIONE:
Tribunale di Milano n. 69 del 20/2/1982

Publicità inferiore al 70%
Spedizione in abbonamento postale Gruppo III/70
Prezzo della rivista L. 4.500. Numero arretrato L. 9.000
Abbonamento annuo (11 numeri) L. 39.000; per l'Estero L. 78.000
I versamenti vanno indirizzati a: Gruppo Editoriale Jackson - Via Rosellini, 12
20124 Milano - mediante emissione di assegno bancario, cartolina vaglia
o utilizzando il C/C postale numero 11666203.
Per i cambi di indirizzo, indicare, oltre naturalmente al nuovo, anche l'indirizzo precedente, ed allegare alla comunicazione l'importo di L. 500, anche in francobolli.

© TUTTI I DIRITTI DI RIPRODUZIONE O TRADUZIONE
DEGLI ARTICOLI PUBBLICATI SONO RISERVATI

Il Gruppo Editoriale Jackson è iscritto nel Registro nazionale della Stampa
al n. 117 - Vol. 2 - Foglio 129 in data 17-8-1982



GRUPPO EDITORIALE JACKSON
DIVISIONE PERIODICI

Milano-San Francisco-Londra-Madrid



Periodico iscritto all'Unione stampa periodica italiana

DIREZIONE, REDAZIONE, AMMINISTRAZIONE:

Via Rosellini, 12 - 20124 Milano - Telefoni: (02) 68.80.951/2/3/4/5

SEDE LEGALE: Via G. Pozzone, 5 - 20121 Milano

DIREZIONE EDITORIALE: Daniele Comboni

DIREZIONE AMMINISTRATIVA: Giuliano Di Chiano

DIREZIONE DIVISIONE LIBRI E GRANDI OPERE: Roberto Pancaldi

PIANIFICAZIONE STRATEGICA: Sergio Mello-Grand

CHIEDI IN EDICOLA IL TUO



C-64 & C-128



MSX



C-16

VIC-20



Spectrum



GRUPPO EDITORIALE JACKSON
DIVISIONE GRANDI OPERE

Suonare è facile.

MUSIC MASTER trasforma il tuo C16 in un fantastico strumento musicale. Anzi, ancora di più: puoi scegliere tra il dolce e melanconico flauto, o il più sfrontato e moderno synt o ancora il simpatico ed essenziale xilofono. Carica il programma SINTETIZZATORE e sperimenta, utilizzando la apposita tastiera, le grandi possibilità del tuo piccolo, grande computer. Le pagine seguenti ti informeranno su come

ciò sia possibile, ti ricorderanno i principi fondamentali della musica. Qualunque sia la tua preparazione in questa materia, sarai in grado di leggere e riprodurre col tuo C16 qualsiasi brano musicale. Grazie al programma MUSIC EDITOR potrai infatti introdurre nella memoria del computer la musica composta da te o ricavata da spartiti, per poi riascoltarla a tuo piacimento.

MUSIC MASTER C16

ED È SUBITO MUSICA CON IL C16

UNICO FACILE, COMPLETO PERCHÉ HAI:

- TASTIERA PER C16
- MANUALE MUSICALE
- PROGRAMMI SOFTWARE

MUSIC MASTER

- ELEMENTI FONDAMENTALI DELLA MUSICA
- IL SINTETIZZATORE
- MUSIC EDITOR
- ALBUM: 4 BRANI DI MUSICA CLASSICA E MODERNA

MUSIC MASTER

- MUSIC EDITOR
- ELEMENTI FONDAMENTALI DELLA MUSICA
- ALBUM: 4 BRANI DI MUSICA CLASSICA E MODERNA

CORSO RAPIDO DI MUSICA A SOLE L.15.000

NUMERO UNICO

commodore **C16**

GRUPPO EDITORIALE JACKSON

È in edicola